Departamento de Automática, Ingeniería Electrónica e Informática Industrial
UNIVERSIDAD POLITÉCNICA DE MADRID

A Foundation for Perception in Autonomous Systems

Tesis Doctoral

Ing. Ignacio López Paniagua
DIRECTOR: Dr. Ing. Ricardo Sanz Bravo

2007

Tribunal nombrado por el Magfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día 13 de abril de 2007.

Presidente: D. Ramón Galán López
Vocal: D. Miguel Segarra Martínez
Vocal: D. Tom Ziemke
Vocal: D. Riccardo Manzotti
Secretario: D. Félix Moreno González
Suplente: D. Manuel Rodríguez
Suplente: D. David Travieso García

Realizado el acto de defensa y lectura de la tesis el día 17 de mayo de 2007 en la E.T.S. Ingenieros Industriales.

CALIFICACIÓN:

EL PRESIDENTE LOS VOCALES

EL SECRETARIO

BINGHAMTON
UNIVERSITY

GEORGE J. KLIR, DISTINGUISHED PROFESSOR
DEPARTMENT OF SYSTEMS SCIENCE AND INDUSTRIAL ENGINEERING
THOMAS J. WATSON SCHOOL OF ENGINEERING AND APPLIED SCIENCE

PO Box 6000
Binghamton, New York 13902-6000
607-777-6509

To: Whom it May Concern
From: George J. Klir
Subject: Ph.D. Thesis entitled "A Foundation of Perception in Autonomous Systems" by Ignacio López Paniagua
Date: January 28, 2007

In spite of the impressive advances in capabilities of machines, they are still not able to match some capabilities of human beings. The most exemplary of them are the powerful perceptual capabilities of the human mind, which allow humans to use perceptions in purposeful ways to perform complex tasks. To emulate these capabilities in machines is contingent upon our understanding of the nature of perceptions. This dissertation is a contribution in this regard. The author has throughout the whole thesis applied ideas emerging from systems science, in particular the notion of autonomous systems, to the study of perception. This approach to perception has been successful in the sense that it resulted in a sound general framework for conceptualizing perception, which can be readily utilized in a recently emerging area known as "computing with perceptions" [Zadeh, L. A., "From computing with numbers to computing with words — from manipulation of measurements to manipulation with perceptions." *IEEE Trans on Circuits and Systems*, **45**(1), 1999, pp. 105-119].

The thesis is an original contribution to a very difficult subject whose importance has been increasingly recognized within the area of computational intelligence. It is conceptually sophisticated, logically consistent, and very well written. I expect that the main utility of the framework for conceptualizing perception developed in the thesis will be in the area of computational intelligence. In summary, the thesis is a respectable scholarly work.

George J. Klir

Universitat de les Illes Balears

Departament de Psicologia

Dr. Antoni Gomila Benejam, profesor titular del área de Psicología Básica del Departamento de Psicología de la Universitat de les Illes Balears,

HAGO CONSTAR:

que el trabajo de investigación doctoral elaborado por el sr. Ignacio López Paniagua, "A Foundation for Perception in Autonomous Systems", cumple sobradamente los requisitos para ser defendido como tesis doctoral.

Este trabajo constituye un ambicioso y sorprendente "tour de force", para desarrollar un marco conceptual apropiado para el estudio de la percepción en los sistemas cognitivos, independientemente de su naturaleza (biológica, computacional,...). Una aproximación análoga sí se ha planteado para los procesos estrictamente cognitivos –dentro de lo que se ha dado en llamar la "epistemología androide", los procesos de representación del conocimiento e inferencia en sistemas cognitivos, al margen de su constitución material-, pero el trabajo de Ignacio constituye la realización de un proyecto parejo en el ámbito de los procesos perceptivos. Hasta el momento, el campo de la visión artificial ha desarrollado sus propios métodos y enfoques, sin buscar tal grado de generalidad y comunalidad. En este sentido, este trabajo supone un esfuerzo pionero, que abre una nueva perspectiva para aprovechar los avances habidos en el estudio de los sistemas perceptivos naturales en el campo de los sistemas artificiales.

Por todo ello, recomiendo la autorización para que este trabajo pueda ser defendido como tesis doctoral.

Palma de Mallorca, 15 de febrero de 2007

Fdo.: Antoni Gomila

Universitat de les Illes Balears
Departament de Psicologia

Edifici Guillem Cifre de Colonya
Campus universitari
Cra. de Valldemossa, km 7.5
07122 Palma (Illes Balears)

Tel.: 971 17 30 00 / 01
Telefax: 971 17 31 90

UNIVERSITY OF SKÖVDE

To whom it may concern

I have read Ignacio López Paniagua's doctoral dissertation *"A Foundation for Perception in Autonomous Systems"* (draft version of February 1). My overall evaluation is that the dissertation is of good quality and makes several relevant contributions to autonomous systems theory. I particularly appreciated the effort to relate and integrate theories from different scientific fields in a systems-theoretical perspective.

Ignacio's PhD research has also already made a several important contributions to the European integrated project *"Integrating Cognition, Emotion and Autonomy"* (IST-027819, FP6) which I coordinate, in particular to the formulation of a theoretical framework for research groups from a range of different disciplines.

If you need any further information please do not hesitate to contact me.

Skövde, 5 February 2007

Tom Ziemke, PhD
Professor of Cognitive Science
School of Humanities and Informatics
University of Skövde, Sweden

+46-500-448330 (phone) / -448399 (fax)
tom.ziemke@his.se

MAILING ADDRESS
P.O. Box 408
541 28 Skövde
Sweden
TELEPHONE
+46 500-44 80 00
TELEFAX
+46 500-41 63 25

This work has been partially supported
by the European Commission through grant

IST 027819 ICEA

v. May 3, 2007

To my two Lauras
To my parents
To my grandmother Julia
To the memory of my lost grandparents.

A mis dos Lauras
A mis padres
A mi abuela Julia
A la memoria de mis abuelos.

Contents

Preface

The studies and reflections about perception are as old as man. Perception is the way through which we obtain much of the information we interpret, alter or distort in order to make our reflections and our decisions, but also to hold objects in our hands, to touch and to move, giving us knowledge about the readiest and also about the most elaborate of realities.

The way in which we perceive the same part of reality may be different each time, in spite that sometimes artifacts, neutral witnesses of our individuality, would measure the same stimuli reaching us from the outside... We all may remember *seeing things with different eyes* before and after some event, happy or sad, has happened to us, or before and after having learned about the physics of matter, or at some time when our attention was not fully focused on the outer world, and more abandoned to our own thoughts.

If sadness, thought and preoccupation, affect our perception, have we any ground to think that other subtler aspects of ourselves do not influence it? How could we be sure that being taller or shorter, or slimmer or fatter would not alter our perception and the reality derived from it? If perception may change inside ourselves, how could we imagine what it is in others? There is a reality outside us which we can perceive but we cannot know... perception is an extraordinary form of blindness.

Through blindness man sees an environment that he sometimes masters, and that he changes to his own convenience convincing him of his own autonomy. There must be a link between environment and man that explains perception, however hidden it may be.

The problem of perception is known to us from many views and through many riddles. Indeed we have all reflected on different waters flowing through the same river (Heracleitus), on the Myth of the Cavern (Plato), on the demolishing uncertainty of knowing but the existence of ourselves (Descartes).

When we try to commence a research on perception we might not be conscious that it is through our own perception that we are going to advance, and that *we intend perception to perceive perception*. Being aware of this circumstance, however, would help us little for other than casting a veil of uncertainty over our thought, for objectiveness and subjectiveness will surely be undistinguishable in our results. In which way can we hold to the scientific method?

Perception, or the aspects that interest man, are far from the tangible and measurable qualities of the environment. Man needs intensity of light to see, and surely man's perception is affected by it, but the desire of man is far from wavelengths and photons and closer to happiness, sadness and beauty. And man perceives them… but through which paths can wavelengths and photons become beauty? How does perception happen?

Man has an obsession for man. Perhaps man is the most perceived part of reality… by man. Yet, what is man? Perhaps others are a reflection, an instantiation of *I*. And what is *I*?

The notion of self is controversial. One thinks of *I* as a set of grouped parts, perfectly separable and distinguishable from the rest. And, being distinguishable and separable, one expects to perceive them as such.

Perhaps *I* is not either clearly distinguishable or separable. Perhaps the self only exists as the subject of our consciousness and the main character of our own soliloquy. Our soliloquy changes and our consciousness as well, as we come to realize as we walk into maturity. We should expect the *I* to change with them. And indeed we read from time to time that we consider our bikes as part of ourselves when we ride, and about the psychological frontier of the self… So what is an *I* that changes driven by a soliloquy and that exists in a tale?

However, if the *I* changes, perhaps we could get hold of reality. And, yet, how are we sure that there is reality outside?[1] We cannot be sure, but let us pretend there is, and write down a thesis to explain how we perceive its beauty.

Madrid, Monday January 22$^{\text{nd}}$, 2007.

[1] *cogito ergo sum*.

Part I

Versión Española

Fundamentos de la Percepción en Sistemas Autónomos

Agradecimientos

Entiendo que uno siempre agradece algo intencionado, concreto y finito en el tiempo. No creo que se pueda agradecer a alguien su forma de ver el mundo, la energía que desprende o el universo que contagia con su persona. Son esas las cosas que más me han ayudado y que me han acompañado desde que inicié el camino que termina ahora. Y aunque su compañía en este tiempo se refleje en un texto como éste, que es algo tangible y concreto, no concluye aquí, puesto que siguen conmigo y conmigo las siento. No podría ser como soy sin ellas. Por tanto, en mi limitada condición, son infinitas y se mezclan con el resto de mí de tal modo que no sé distinguirlas de mí mismo.

Por todas estas cosas, no agradeceré aquí las cosas que tengo que agradecer. Dedicaré el capítulo a reflexionar sobre esas cosas infinitas que tanto valoro, y a recordar en palabras a algunas de las personas en las que nacieron o se inspiraron métodos de trabajo y de pensamiento que ahora son como míos, e ideas escritas aquí que también ahora son como si fueran mías.

Una frase, por su maravillosa precisión y porque siempre me ha parecido un testimonio de humildad inteligente y verdadera, me mueve a escribir este capítulo. *Si he logrado ver más lejos, ha sido porque he subido a hombros de gigantes.* Parece ser que Isaac Newton se la escribió a Robert Hooke en 1676.[2] Yo no creo que mi vista haya llegado más allá de medio paso, y por supuesto no creo que haya visto ni la infinitésima parte de lo que vio la de Newton, a pesar de la posición ventajosa que me ha dado el tiempo. Pero sí siento que mis días, mis pensamientos y mi vida estarían mucho más cerca del vacío o ahogados en él si no fuera porque tengo la fortuna de caminar cerca de gigantes.

Pensaré, antes de nada, en el *pensamiento*. Casi todos los que emprenden un camino como éste dieron el primer paso en un tiempo que prácticamente no recuerdan ya, de tan lejano. Cuando aún eran niños y esa fascinación intensa por la vida y las cosas del mundo aún inundaba sus pensamientos y abría sus ojos. Cuando todo era objeto de curiosidad y las preguntas eran insaciables. Eso me pasó a mí.

Después, ya nublándose la despierta fascinación, empezando la carrera, para muchos llegó la luz de los grandes: Newton, Gauss, Euler, Einstein… Y después muchos

[2] *A Hombros de Gigantes: Las grandes obras de la física y la astronomía*, edición comentada de Stephen Hawking. Ed. Crítica. Cuarta edición, febrero de 2005.

los olvidaron y hoy parece que no les dejaron nada. Eso no me ocurrió a mí. Es el mismo espíritu el que me ha llevado en todos los momentos, hasta hoy.

Mientras pensaba las cosas que ahora he escrito, recibí muchos más rayos de luz. Es algo que me gusta recordar. Los espléndidos ratos leyendo a Herbert Simon, a Allen Newell y a Irvin Rock; hubo muchos más. Recuerdo especialmente el momento en que abrí por primera vez el libro que quizá más me impactó durante todo este tiempo, y en el que se basa este texto —aunque yo entonces estaba muy lejos de sospechar que eso llegaría a ocurrir— *An Approach to General Systems Theory*, de George J. Klir. Se me antojó tan inmenso aquella primera vez, que lo abandoné un tiempo durante el que viví con la intuición de que encerraba, espléndidamente claras y magníficamente estructuradas, las marañas de ideas sobre sistemas que empezaban a bullir en mi cabeza. Termino este párrafo ahora, al final, cuando hace dos días que recibí un informe del autor del libro refiriéndose a este trabajo, que ha tenido la delicadeza de leer y valorar. Es para mí un honor y motivo de alegría inmensa. Me hace pensar que verdaderamente he estado caminando por la senda eterna de la ciencia, herencia que generaciones de maestros han dejado a sus pupilos y por la que anduvieron antes que yo todos los gigantes desde los tiempos remotos.

Tom Ziemke, Hans Georg Stork y Toni Gomila tuvieron la amabilidad de acceder a revisar este trabajo; me sorprendió su actitud y su fantástica disposición. Imagino que todavía estarán leyendo y espero que después de hacerlo decidan repetir con otro alumno que se lo pida alguna vez.

Ahora olvidaré el pensamiento, porque no es más que la introducción a los recuerdos a los que dedico este capítulo, y pierde el interés. Yo sé que mi pensamiento es seguro, rápido y demoledor, siempre que le ilumine mi luz más brillante. Esto, en alguien como yo, solamente ocurre en ratos perdidos en el tiempo, porque alguien como yo vive entre luces y sombras. A veces hay muchos años de sombras sin un atisbo de luz de consuelo, y yo he vivido muchos años de oscuridad terrible. En ellos no hay deseo, ni espíritu; ni nada. Y el recuerdo de la luz de aquellos que iluminaron antaño se antoja vacío, ajeno e insignificante. *Abandonad toda esperanza.*

Esa oscuridad es impenetrable. El camino a la luz es incierto, y muchos no vuelven a ella jamás. Hay que creer ciegamente que se volverá a ver y obligarse a presentir claridad donde en realidad solo sigue habiendo el mismo dolor ahogado y negro en el que uno vive.

El tiempo se pasa dando pasos erráticos sin saber por qué y sin que lleven a ninguna parte; a veces se para y se agoniza quieto durante tiempos infinitos que acaban en nuevos pasos doloridos hacia ningún lugar. A base de engaños y voluntad cuando ya no queda voluntad ninguna, puede uno maldecir al destino y hacer que uno de entre esa infinidad de pasos sea el primero hacia una claridad aún más anhelada que sentida.

Guardo infinidad de recuerdos y siento admiración infinita hacia algunas personas que he presentido entre sombras en muchos momentos difíciles. Me llena de alegría y entusiasmo verles claramente a la luz. Admiro de muchos su tesón, de otros su templanza, de otros su paciencia, de otros su prudencia, de casi todos su bondad y de todos ellos, de una u otra manera y sometida a todas las distorsiones del ser humano, su virtud.

Entre los que no he conocido, no puedo olvidarme de Alfredo Kraus, a quien mi pensamiento ha tratado de imitar en su inteligencia, y quien ha interpretado todos los colores de mis emociones. Incontables veces, con él o a través de él, Giuseppe Verdi; maestro del pensamiento y el sentimiento del alma humana. Envidio la serenidad inteligente de Clint Eastwood que quisiera compartir y que intento copiar muchas veces.

Pero también he tenido la suerte de conocer a personas excepcionales y de que lo hayan sido conmigo. No quiero olvidarme de Fernando Martín Santos, Jesús Portal, Pepe Rodríguez —a quien no sabría cómo encontrar—, José Manuel y de la memoria de Sonia. Todos ellos, sin saberlo, fueron y son ejemplo y sabiduría para mí.

Por supuesto, recuerdo a Jaime, con quien he podido pasar algunos ratos en estos últimos meses, y a quien quiero pedir que piense: *Que tiene el DEBER de hacer cuanto pueda por volver esta realidad en la realidad que los dos hemos soñado tantas veces; sin excusas; sin caer en el desánimo. Y sobre todas las demás cosas, sintiéndose heredero de los grandes y no ahogado por la vulgaridad a cuya altura no debemos dejarnos caer nunca. Cajal, Betancourt, de la Cierva, Ricart, y naturalmente, su querido Ortega, no nos deben servir solo para saber lo que se debe hacer y no se hace, sino, sobre todo, para ver en ellos que se puede hacer lo que parecía imposible. Y yo creo que ahora es más posible que entonces... a pesar de todo.*

Recuerdo también a los amigos con los que empecé la carrera, aún en los tiempos de candidez: Jesús, Celia, Marisa, las dos Anas, Almudena, Isadora, Óscar, Miguel, Rafa, Carlos, Carmen, Mariola, Juan Ignacio, José Manuel, José Luis, Jorge... Sigo viendo a la mayoría de ellos al menos una vez al año; aunque este haya fallado precisamente por estar día y noche encerrado escribiendo este trabajo.

He compartido momentos maravillosos con muchísimos más amigos, a los que admiro profundamente aunque a veces no hayan podido ni imaginarlo. Fernando, Pablo, Inma, Ana, Asís, Enrique, Merche, David, Álvaro, Miguel, Rafa, Luis, Javi, Alberto, Dani. Algunos de nosotros llevamos vigilándonos más de veinte años, desde cuando veinte años todavía nos parecía algo muy largo entre pinos, rocas y montañas.

Y a mis compañeros hasta hace muy poco; a los que deseo lo mejor del futuro solo si pelean por ello abandonándose a la rabia, porque solo así acabarán con el universo viscoso que les oprime. Félix, Javier, Fernando y Óscar.

Con ellos he compartido ratos y épocas enteras de mi vida. Espero pasar una eternidad intentando hacer mías un poco de la comprensión y la sensibilidad de mi novia Laura, y escucharla al piano...

[*rall.*] e qui la [*f*] lu-na, l'ab-[*affrett.*]-bia-mo [*pp*] vi-[*poco rit.*]-ci-na...

¿Laura, lees?

Compartiré esa misma eternidad con mi hermana, maldiciendo del mundo, padeciéndolo, y de vez en cuando riéndonos de él entre tés, gestos histriónicos y paseos bohemios y solitarios. Y compartiré el momento de defender esta tesis con mi abuela, aunque su timidez le haga temblar un poco.

Quisiera recordar expresamente a Ricardo, aunque me gustaría que él supiese que estaría presente en este trabajo aunque no hubiese escrito su nombre aquí. No solo quiero recordarle, también quiero que, si alguien leyese estas líneas, supiese quién es Ricardo.

Desde el punto de vista de capacidad científica o técnica debería haberlo mencionado antes, cuando hablé del pensamiento. Pero no lo he hecho porque hace mucho tiempo que no lo veo en ese mundo.

Ricardo es un investigador humanamente excepcional y extraordinariamente generoso. Encarna el idealismo y el espíritu ilustrado de ciencia que en otro tiempo fue el que llevó a los hombres a realizarse como hombres a través de la pasión humana por el conocimiento.

No hay alumno malo ni alumno que no pueda venir a trabajar con él, ni alumno ni persona que no pueda venir a preguntarle. De los que estamos con él no hay quien no le admire.

Su implicación con las personas que están a su alrededor es impresionante y muchas veces conmovedora, y lo ha sido para mí, personalmente, en tantas ocasiones que no las podría contar. Incluso en circunstancias que fueron mucho más desagradables en la realidad de lo que me atrevo a recordarlas.

De todas estas cosas no puedo decir lo mismo de casi nadie en la Universidad. Yo las interpreto como una muestra de humanismo auténtico y de humildad consciente y elegida para con el ser humano y con la ciencia.

No puedo decir que alguna vez haya dudado de terminar esta tesis doctoral, porque no es cierto. Pero sí hubo un tiempo en el que no supe qué era y en que apenas supe lo que era mi vida. Por supuesto, ha habido momentos en los que el final se antojaba infinitamente lejano. Bien, hoy está tan terminada como puede estarlo un trabajo de investigación, es decir, apenas empezado. Ricardo, una vez más, me ha ayudado a través de una dirección dedicada y, en mi opinión, experta. Es cuanto todos hubiesen esperado de un Director. Pero estas cosas son las que prefiero no desarrollar aquí, para dejar toda la importancia al ejemplo y el estímulo que supone para mí por todo lo demás.

Finalmente voy a dedicar las últimas líneas de este capítulo a pensar en mis padres. En contadas ocasiones me veo tentado de ordenar mis pensamientos sobre ellos en un lenguaje comprensible para el resto del universo. Y no me suele gustar hacerlo, porque se me alcanza demasiado pobre.

Hace muchos años que tengo a mis padres por gigantes a cuya sombra aprendo y cuya sombra sigo. Lo digo porque dejaron de ser *mis padres* cuando comprendí que vería su inmensidad aunque no lo hubiesen sido.

Se juntan en ellos infinidad de virtudes que admiro en abundancia infinita. Abnegación, comprensión, paciencia, responsabilidad, bondad, inteligencia, humildad. No es por casualidad o exclusivamente por naturaleza, sino por convencimiento y esfuerzo de superación humanos, lo que me las hace aún más dignas de admiración. Y además tienen, igual que siempre han tenido, la grandeza de regalarlas a quien las necesita.

No soy capaz de concebir mis días sin ellos, ni mis sentidos ni mis pensamientos. En todos los razonamientos que he seguido en este trabajo les veo a ellos pensando; en los enfoques que he diseñado les reconozco a ellos; muchas situaciones que me han inspirado las viví con ellos. Es inevitable que cuando leo mi nombre en la portada lea los suyos entre las letras del mío.

Ignacio López
Madrid, tarde del domingo 4 de febrero de 2007.

Introducción, objetivos y comentarios preliminares

Esta tesis doctoral ofrece una visión conceptual de la percepción, con la intención de ser aplicada en el futuro al análisis y al diseño de sistemas artificiales. Tal y como está expuesta en este trabajo, ofrece un marco conceptual que permite extraer principios y reglas generales sobre el funcionamiento de los sistemas. El marco construido no está formalizado, por lo que aún no permite una cuantificación. La formalización y la cuantificación son dos pasos que deberían seguir a este trabajo >*cap.3*.

Se observará que la tesis tiene dos capítulos. El primero, cap. 1, está dedicado a los sistemas autónomos. Tiene por objeto explicar conceptos abstractos tales como la autonomía, y cómo se relacionan con la operación y la estructura de los sistemas. Se comienza aportando una visión de los sistemas artificiales existentes desarrollada en torno al concepto de autonomía. A continuación, se exponen los conceptos más importantes de la teoría general de sistemas tal y como se entenderá en este trabajo. A partir de aquí, se desarrolla una visión de los sistemas autónomos y generales que trata de integrar aspectos internos (como su estructura) con otros externos (como su autonomía o su comportamiento). Este contexto determina el papel de la percepción en los sistemas autónomos y la forma en que tiene lugar. Ilustra las relaciones de la percepción con los demás aspectos sistémicos y las restricciones potenciales a las que está sometida.

El segundo capítulo (cap. 2) está dedicado a la percepción propiamente dicha. Como se decía, se entiende que la mayoría de sus aspectos estructurales se derivan directamente del contexto sistémico desarrollado en el capítulo anterior. En línea con esta idea, este capítulo analiza la percepción desde dos puntos de vista que derivan del concepto de sistema: Por una parte, las partes que intervienen en la percepción y la manera en que están relacionadas. Por otra parte, los flujos de información en el sistema.

Finalmente, en el capítulo 3 se incluye una exposición con cierto nivel de profundidad explicando las principales conclusiones y líneas de progreso previstas para este trabajo.

La versión española de esta tesis es un resumen de la versión inglesa. Es esta última la que debe consultarse. La versión española está construida sintetizando la mayoría de conceptos y las explicaciones de la inglesa. Secciones enteras han sido excluidas del resumen, así como referencias, figuras y ejemplos. Sin embargo, la estructura de la versión inglesa se ha respetado con el fin de facilitar al lector referirse a los contenidos de esta versión.

Los objetivos principales de esta tesis son los siguientes:

Generalidad: Explicar la percepción desde un punto de vista general, estableciendo una ontología común para sistemas artificiales y biológicos.

Obtener conceptos, principios y relaciones aplicables al diseño de sistemas artificiales. Este objetivo incluía una formalización de la ontología.

Estos objetivos se formularon desde el convencimiento de que los niveles de complejidad y la naturaleza de las tareas de los sistemas artificiales actuales exceden los niveles de prestaciones ofrecidos por la ingeniería convencional. La generalidad, eventualmente, permitirá diseñar soluciones bioinspiradas eficientes a problemas todavía sin resolver.

El diseño bioinspirado ha existido en la ingeniería desde tiempos remotos. Ejemplos de aproximaciones recientes de este tipo pueden encontrarse en arquitecturas cognitivas conocidas como RCS y SOAR.

Este trabajo se enmarca en una línea de investigación centrada principalmente en la investigación sobre teorías y principios generales más que sobre problemáticas y aplicaciones concretas, aunque éstas formen parte de ella necesariamente. Su foco es la ingeniería del conocimiento, aplicada a cualquier faceta del diseño de sistemas. La generalidad es una condición necesaria para ésto.

El interés por la formulación general de problemas no es nuevo. Su expresión más clara se dio con el nacimiento de la teoría general de sistemas, *general system theory*, GST a mediados del siglo XX. De hecho, este trabajo se basa en conceptos sobre sistemas heredados de una de las formulaciones de esta teoría, recogida en el libro *An Approach to General Systems Theory*, por George J. Klir [Kli69].[3]

El grado en que se ha alcanzado los objetivos expuestos arriba se discutirá en un capítulo al final de esta versión española, y más en detalle en su homólogo de la versión inglesa. Sin embargo, es conveniente avanzar que no se ha completado al formalización y que solo se ha alcanzado un grado elemental.

La metodología diseñada para llevar a cabo este trabajo se ha basado en el ideal del método científico tradicional, que podemos resumir en un ciclo de tres fases fundamentales: (1) experimentación (2) observación y (3) generalización. De acuerdo con este ideal, la experimentación sirve tanto como punto de partida como referente frente al que comprobar las nuevas teorías.

Debido al alto grado de multidisciplinaridad de este trabajo y el interés por la generalidad, fue necesario reinterpretar el ideal para hacer la investigación posible con tiempo y recursos limitados.

[3] Versión española: *Teoría General de Sistemas* [Kli80].

La fase de 'experimentación' se reformuló en un análisis experimental, en en cual se analizó tanto la experiencia previa del Grupo en sistemas inteligentes de control como multitud de fuentes externas al grupo, y de diversas áreas. Llevar a cabo experimentos de psicología, neurociencia, ingeniería, geometría y todas las demás disciplinas en que se ha basado este trabajo hubiera sido imposible. Se ha realizado un gran esfuerzo para extraer principios generales de los trabajos anteriores del Grupo y de literatura científica, que pudiesen aportar casuística de referencia y cumplir el mismo papel que la experimentación propiamente dicha.

Esta interpretación del método científico, adoptada al principio de la investigación, la hemos entendido como esencial más tarde, a medida que sumábamos más áreas de conocimiento a aquellas de las que partimos: percepción en sistemas biológicos.

A la hora de poner por escrito el trabajo realizado en los útlimos años, ha sido necesario restringir el punto de vista del discurso para hacerlo más comprensible y darle coherencia. Por ello, se ha omitido diversos temas que fueron importantes para llegar a la conceptualización propuesta aquí. Entre ellos se cuentan la *consciencia*, los *sistemas paralelos distribuidos*, los sistemas *de tiempo real*, estudios sobre *arquitecturas cognitivas*, y otras disciplinas, en menor grado, como la *geometría*.

Es conveniente señalar, por último, que la mayoría de los conceptos expuestos aquí, de acuerdo con los objetivos expuestos arriba, son generales y tienen un alto nivel de abstracción. Por tanto, deben explicar tanto lo complejo como lo simple, lo concreto y lo abstracto, lo artificial y lo biológico de una manera coherente. En los sistemas reales, muchos de los conceptos que aquí se mencionan pueden no aparecer, o hacerlo de una forma muy primitiva, mientras que otros pueden darse de forma muy desarrollada. No existe sistema alguno conocido por el autor que desarrolle plenamente todos los conceptos expuestos aquí.

Chapter 1

Sistemas autónomos

1.1 Estado del arte de los sistemas autónomos

Los sistemas autónomos vienen siendo estudiados en ingeniería desde el comienzo de su historia. Un sistema autónomo se entiende como aquél que es capaz de operar sin intervención humana [HMH04], [WJ94], [Ken03]. Esta noción deriva de *autos* (uno mismo) y *nomos* (ley) [Bat01, p.118]. Por tanto hablar de sistemas autónomos es hablar de teoría de control, automatización y robótica.

La razón de que nosotros empleemos el término *sistema autónomo* en esta memoria es doble. Por un lado, queremos distinguir el enfoque desde el cual los estudiamos, que tiene una fuerte componente teórica. Por otro lado, queremos recalcar que nuestra investigación pretende tratar sistemas más allá del estado del arte: sistemas de muy alto grado de autonomía, es decir: comparable al de los humanos o al de algunos animales.

1.1.1 Tipos de autonomía

Esencialmente, se distingue dos tipos [GL04, p.2]:

(a) Operacional: la capacidad de un sistema para compensar perturbaciones inducidas por el entorno.

(b) De decisión: la capacidad de un sistema para tomar sus propias decisiones.

Esta es una clasificación primaria de los tipos de autonomía. Clasificaciones más detalladas han sido propuestas y analizadas desde distintos puntos de vista en diversas fuentes: [GL04], [Mey00], [HMH04].

1.1.2 Los sistemas para el ingeniero

Para un ingeniero, un sistema es una cierta cantidad de variables que pueden ser de tres tipos: las entradas, las salidas, y las variables internas. Las entradas representan el medio por el cual el humano o el entorno interaccionan con el sistema. Las salidas son los parámetros que actúan sobre el entorno y que nosotros observamos (junto con las variables internas) como ingenieros, para comprobar que el sistema se comporta de forma adecuada. Esta noción subyace a toda teoría de sistemas, control o automatización, y la llamaremos noción general de sistema.

1.1.3 La autonomía en sistemas reales: los problemas a resolver

Diseñar sistemas para operar autónomamente en condiciones reales conlleva varias dificultades. Podemos clasificarlas en las siguientes categorías:

1. **Perturbaciones:** Cuando un ingeniero diseña un sistema para que éste haga algo concreto (que se le comunica a través de sus entradas), lo hace en función de las entradas esperadas. Sin embargo, la interacción entre el entorno y el sistema también se produce mediante entradas no esperadas o perturbaciones. Algunas de estas perturbaciones (el tipo de las mismas) pueden ser conocidas de antemano, pero otras son completamente desconocidas.

   El diseño del sistema realizado por el ingeniero está hecho considerando condiciones normales de operación del sistemas, es decir, que el sistema recibe (en condiciones normales) únicamente las entradas esperadas. Pero la realidad es que también el entorno puede afectar (y afecta) al comportamiento del sistema de formas no esperadas, y para las cuales el sistema no está concebido. La existencia de estas perturbaciones han conducido a que los sistemas se diseñen con mecanismos adicionales que permitan que los mismos sigan funcionando de forma adecuada incluso en condiciones anómalas. Aparece la necesidad de (al menos) una supervisión externa y (la mayoría de las veces) de un sistema compensatorio que realiza una acción externa (el sistema de control). Este es, de forma clara, un primer obstáculo para conseguir sistemas de plena autonomía.

2. **Abstracción:** El funcionamiento de los artefactos se ha basado tradicionalmente en parámetros bien determinados, y según leyes formalizables matemáticamente. Así, los sistemas automáticos tradicionales, se basan en control por realimentación. La complejidad de estos sistemas, como en el caso de aplicaciones industriales como las plantas de producción de energía o de proceso químico, se ha incrementado haciendo que sistemas de control sean controlados y/o supervisados a su vez por otros sistemas de control. Los sistemas que controlan otros sistemas se suele decir que operan a mayor nivel de abstracción, porque en general controlan el cumplimiento de objetivos a más largo plazo que los sistemas simples.

   Sin embargo, la abstracción ha aparecido en los sistemas artificiales de otra manera en los últimos años, al incrementarse los requisitos de interacción con humanos. Cada vez más, se desea que los sistemas se muevan autónomamente en entornos no controlados: ferias, congresos, fábricas, hospitales, y se desea que muestren un comportamiento socialmente agradable. Estos requisitos sirven para darse cuenta que los sistemas capaces de ejecutar estas tareas deben operar de alguna manera con variables que no se pueden medir tal y como se hacía en el control tradicional.

A este tipo de variables, como pueden ser el enfado de los visitantes de una feria al toparse con un robot, se les llama abstractas.

3. Incertidumbre: El último gran problema al que se enfrenta el ingeniero para diseñar sistemas autónomos es el no saber exactamente las condiciones en que va a tener que operar el diseño. En ciertos dominios no se dispone de 'unas condiciones normales de operación' para la realización del diseño. En efecto, si no se controla el entorno de operación de un sistema, muchos factores pueden ocurrir de forma imprevista; tal vez el ejemplo más característico sea la aparición de obstáculos, como un peatón que se cruzase por delante de un robot en movimiento. La incertidumbre del entorno implica tener que reaccionar ante lo desconocido.

1.1.4 Los mecanismos para lograr autonomía

Arquitecturas de control: El mecanismo de control por realimentación, ya mencionado arriba, constituye la base de la ingeniería de control tradicional. Se trata de comparar el valor de la variable de salida del sistema con el valor deseado, y modificar las entradas de forma que se corrija la diferencia entre ambas. El ejemplo de controlador más representativo es el proporcional-integral-diferencial: PID, el más utilizado en la industria. La evolución natural de este tipo de sistemas ha seguido el paradigma de la pirámide de control, por el cual los controladores más simples, PID, son a su vez controlados por dispositivos programables: programmable-logic controller PLC, y éstos a su vez coordinados por ordenadores. De esta forma, el control de una variable en una planta (temperatura de una sala, velocidad de una fresadora, etc.) es escalado al del grupo completo (una cadena de producción formada por varias salas y fresadoras) y éste a su vez al de una fábrica (varias cadenas de producción). El nivel más alto del control correspondería a la planificación para alcanzar unos determinados objetivos de producción. Generalmente, los niveles superiores de control son difícilmente automatizables debido a su complejidad y a la abstracción de los conceptos implicados: líneas estratégicas, etc.

Referencias: [Oga90], [SL91], [Che00], [GL00].

Arquitecturas reactivas y basadas en módulos de comportamiento: Las arquitecturas de control como las anteriores han sido empleadas con éxito en el control industrial durante décadas. Sin embargo, dependen de operar en entornos controlados, en condiciones de incertidumbre limitada. En robótica, área en la que frecuentemente se diseña sistemas para que se muevan solos por su entorno, generalmente no se puede evitar la aparición de obstáculos, o la necesidad de explorar un entorno inicialmente desconocido. Esta circunstancia dio lugar al nacimiento de formas de control capaces de reaccionar a la evolución de su entorno. Se caracterizan porque los sistemas diseñados así, siempre se comportan de la misma forma ante la misma combinación de valores de sus variables de entrada (sensores: infrarrojos, de contacto, láser, etc.) Las arquitecturas basadas en módulos de comportamiento representan una evolución de las anteriores. Ante una combinación de sus entradas, realizan un comportamiento, o secuencia de acciones.

Referencias: [Bro91a], [Ark98].

Arquitecturas guiadas por objetivos: Este tipo de arquitecturas está diseñado específicamente para operar en entornos de alta incertidumbre. Se dice que están guiadas por objetivos debido a que son capaces de reconfigurarse para adaptarse a los cambios del entorno para lograr un objetivo. Generalmente se basan en una secuencia de operación con los siguientes pasos: (a) construir un objetivo (b) analizar el entorno (c) diseñar una tarea para alcanzar el objetivo en las condiciones dadas (d) si el objetivo no fuese alcanzable, descomponerlo en objetivos más simples, y proceder con cada uno siguiendo esta misma secuencia. Este tipo de arquitecturas se utilizan actualmente para control autónomo de vehículos militares, diagnosis médica, y resolución de problemas entre otras.

Referencias: [New90], [WB94], [HR95], [Alb99] , [Alb95], [Alb91], [GMP$^+$01].

1.2 Teoría General de Sistemas

La noción de *sistema* es común en las disciplinas científicas, como un concepto clave para modelizar diferentes tipos de fenómeno, referidos a conjuntos de materia, dispositivos, componentes o, en general, entidades. Sin embargo, la noción de sistema también se utiliza en otros dominios como la sociología o la economía. La Teoría General de Sistemas (General System Theory, GST), más propiamente llamada también Teoría de los Sistemas Generales, surgió bajo la idea de que existe una noción de sistema común a todas las disciplinas, que se conocerá como *sistema general*. Esta noción podría expresarse prescindiendo de los aspectos propios de cada disciplina, y extrayendo los comunes.

1.2.1 Marco histórico de la GST

Históricamente han existido diferentes aproximaciones a la Teoría de los Sistemas Generales, con diferentes orígenes, adoptando distintos puntos de vista. Ludwig von Bertalanffy, aceptado como el pionero en la formulación del concepto de sistema general tal y como lo conocemos hoy, encontró indicios de la incipiente concepción de esa idea ya en el siglo XVII, y trazó su desarrollo hasta hoy a través de los trabajos de los grandes personajes de la historia de la ciencia tales como Leonardo, Descartes y Poincaré. Su obra *General System Theory*, [vB69], representa el punto de partida de la Teoría de los Sistemas Generales. El interés en los aspectos generales, sistémicos, de fenómenos ya conocidos y estudiados había venido creciendo desde principios del siglo XX, y había dado lugar al nacimiento de nuevas perspectivas científicas como la *Teoría Gestalt*, [Köh69, Köh59, Ell97, WD04], cuya filosofía se puede expresar en estos términos (traducción del autor):

> La tesis principal de la Teoría Gestalt puede formularse como sigue: Existen contextos en los que lo que está ocurriendo en el todo no puede ser deducido

> de las características de las partes concebidas aisladamente; sin embargo, lo que está ocurriendo en una de las partes está determinado por las leyes de la estructura interna del todo.[1]

El grupo de personas que han inspirado y aportado ideas a la Teoría de los Sistemas Generales es muy numeroso, y muchos de ellos se encuentran entre los científicos más reconocidos como Zadeh [Zad65] o Shannon [Sha48]. Algunos de ellos propusieron postulados científicos sin precedentes, desde puntos de vista basados en consideraciones filosóficas muy profundas, que llegaron a provocar encendidas polémicas e intensos debates científicos. [Gai78] ofrece un estado del arte comentado de la GST, y visiones más actuales pueden encontrarse en [Boj04] y [Web]. Este texto se basa en una obra concreta: *An Approach to General Systems Theory*, escrito por George J. Klir, [Kli69]. La terminología española propuesta en este texto se ha heredado en parte de la versión española del libro original [Kli80]. Algunos términos se han sustituido con el fin de evitar confusiones debidas a la evolución en el significado de algunos desde la fecha de traducción. Esta aproximación representa una tendencia formal y en algunos aspectos determinista, frente a otras, partidarias de conceptos menos precisos como la encabezada por Zadeh, que dio lugar a la lógica borrosa.

1.2.2 Nociones básicas

Pensemos en lo que entendemos por sistema, considerándolo en relación a lo que le rodea. Si todas las posibles entidades que existen forman el *universo*, podemos decir que un *sistema* es una parte de él, que se considera aisladamente del resto para su investigación. Todo aquello del universo que no forma parte del sistema, se llamará *entorno*. En general, las disciplinas de la ciencia comparten esta noción, aunque con matices diferenciadores, usualmente referidos a los criterios para la separación entre sistema y entorno.

El observador distingue el sistema estudiado del entorno evaluando un conjunto de aspectos que entiende como rasgos característicos del sistema, o *atributos*. Estarán caracterizados por los valores de una serie de *cantidades*. Algunas de estas cantidades serán medibles y se les llamará *cantidades físicas*, como la masa, longitud, densidad, etc. No siempre las cantidades serán medibles, en cuyo caso serán *cantidades abstractas*. Los valores de las cantidades, por tanto, es lo que realmente se observa en el entorno, y lo que el investigador utiliza para evaluar los atributos del sistema.

Los instantes de tiempo y los lugares del espacio donde se observa las cantidades constituyen la *resolución espacio-tiempo*. En la investigación de un sistema, por tanto, se efectuará repetidas observaciones de las cantidades, en determinados puntos del espacio, que tras el período de observación tendrán como resultado un conjunto de valores, al cual se llamará *actividad del sistema*.

[1] Max Wertheimer, *Gestalt Theory*, Social Research 11 (traducción al inglés de la ponencia en la Sociedad Kant, Berlín, 1924).
Fuente: http://www.gestalttheory.net/

Sin embargo, si se trata de explicar el comportamiento de un sistema, disponer de un registro de su actividad en muchas ocasiones no resulta suficiente, ya que existen aspectos que pueden no estar recogidos en ella. Dicho de otro modo, pueden existir cantidades que no han sido observadas, pero que intervienen en el comportamiento del sistema. A estas cantidades se les llama *cantidades internas*, mientras que a las cantidades observadas se les llama *cantidades externas*. Para referirnos al conjunto de valores de las cantidades del sistema en un instante determinado decimos *estado* del sistema. Podríamos distinguir entre *estado interno* y *estado externo* en función de las cantidades.

La función principal del investigador es explicar la actividad del sistema. Para ello el investigador analizará ésta tratando de reconocer patrones entre los valores de las cantidades. Generalmente, estos patrones se expresan en forma de relación entre las cantidades, o de función, en el caso de admitir una formulación matemática. A estas relaciones entre cantidades les llamaremos *relaciones de comportamiento*. El conjunto de todas ellas será formalmente el *comportamiento* del sistema.

Podemos observar que el comportamiento del sistema, o dicho de otro modo, el hecho de que presente unas relaciones u otras, es debido a sus *propiedades*. Llamaremos *organización* del sistema al conjunto de todas sus propiedades.

1.2.3 Definiendo sistemas

El estudio de un sistema como un todo puede resultar una tarea extremadamente difícil, debido a la propia complejidad del sistema o a otros factores como la no-observabilidad de alguna de sus partes. Generalmente, para estudiar sistemas complejos, el conjunto total de cantidades se divide en subconjuntos, y a cada uno se le considera como si fuese un sistema en sí mismo. A cada uno de estos subconjuntos se le llama genéricamente *subsistema*, para expresar que en realidad se concibe como parte de un sistema superior. También se puede considerar que los subsistemas son partes constituyentes del sistema (son, en el fondo, subconjuntos de sus cantidades) en cuyo caso se les llama *elementos*. Al conjunto formado por todos los elementos se le llama *universo del discurso* del sistema.

En general, los elementos de un sistema no constituyen elementos independientes y disjuntos, sino que tienen partes comunes. Es decir, que entre dos elementos puede existir un grupo de cantidades compartidas por ambos, que se llamará *acoplamiento* entre los elementos. Se entiende que puede haber acoplamientos entre más de dos elementos.

Los elementos de un sistema, por tanto, están relacionados entre sí a través de sus acoplamientos, lo cual hace que la actividad de algunos elementos dependa de las de otros. El conjunto de elementos y su jerarquía de acoplamientos, por tanto, definen una estructura, que se conocerá como *estructura del universo del discurso y los acoplamientos*, y se abreviará por *estructura–UC*.

Pero el sistema no queda completamente definido por su estructura–UC, a pesar de que ésta explique las partes que lo componen y cómo están relacionadas entre sí. Es necesario conocer qué combinaciones de los valores de sus cantidades son posibles, es decir, qué estados son posibles para el sistema. Además, es necesario conocer a cuáles de ellos podría evolucionar el sistema, partiendo de uno dado, es decir, las *transiciones* posibles desde cada estado. El conjunto de los estados posibles del sistema y sus transiciones asociadas se conoce como *estructura de estados–transiciones*, y se abrevia por *estructura–ST*. Se puede observar que la estructura–ST representa una estructura de la dinámica del sistema, en cuanto que determina parcialmente cómo éste puede evolucionar.

Podemos observar que las estructuras UC y ST representan la organización del sistema, porque definen las posibilidades del sistema de acuerdo a sus propiedades. Sin embargo, para conocer un sistema completamente es necesario completar el conocimiento de su organización con más aspectos que ya han sido mencionados: la resolución espacio–tiempo, una actividad y al menos, las expresiones fundamentales de su comportamiento. De esta forma quedarían determinados todos los aspectos de un sistema dado en un periodo de tiempo concreto. Específicamente, la definición de un sistema consta de cinco *rasgos primarios*:

- El conjunto de cantidades externas y la resolución espacio–tiempo.
- Una actividad dada.
- *Comportamiento permanente*.
- Estructura–UC *real*.
- Estructura–ST *real*.

Queremos señalar que en esta enumeración se ha empleado los adjetivos 'permanente' y 'real', cuyo significado no se ha explicado aquí por motivos de claridad y brevedad. Intuitivamente, podemos asumir que algunas partes de la organización y del comportamiento de un sistema pueden evolucionar, mientras que otras permanecerán constantes. Es decir, algunas partes de la estructura de un sistema pueden cambiar eventualmente, y algunas pautas de comportamiento también. Los adjetivos 'permanente' y 'real' se refieren a las partes invariables del comportamiento y la estructura del sistema respectivamente.

Analizando los cinco aspectos necesarios para definir un sistema, podemos deducir que los dos primeros lo determinan en un sentido circunstancial, es decir, en un lugar y periodo de tiempo concretos: en un contexto determinado. Por otra parte, los últimos tres aspectos se refieren a las características intrínsecas y constituyentes del sistema, que son independientes de la coyuntura en que se encuentre.

También podemos deducir que definiendo un sistema a través de los cinco rasgos primarios conseguimos determinarlo perfectamente, como hemos mencionado, pero que podemos definirlo parcialmente si especificamos solo algunos de ellos. En este caso, nuestra definición no determinaría un solo sistema, sino que definiría un conjunto de sistemas que se ajustarían a los rasgos primarios de nuestra definición. Al conjunto de sistemas definido por uno o más de los rasgos primarios se llama *clase de sistemas*. En muchas ocasiones los investigadores no están interesados en un sistema en particular, sino en una clase de sistemas que tienen rasgos iguales.

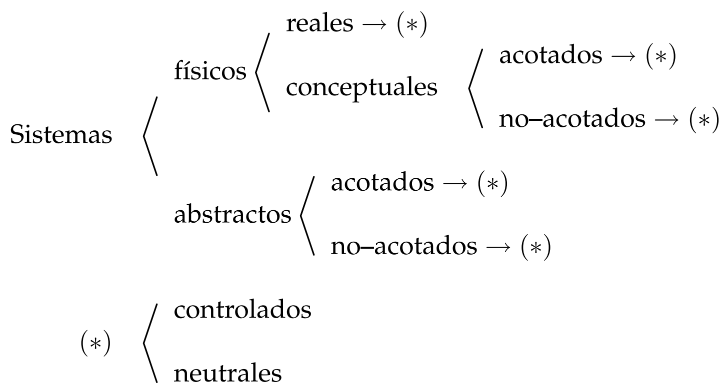


Figure 1.1: Clasificación resumida de los sistemas.

1.2.4 Clasificación de sistemas

Los conceptos de cantidad y estructura introducidos en las secciones anteriores pueden dar lugar a una clasificación de los sistemas. Podemos considerar la clasificación de los sistemas que se ofrece en la figura 1.2.4. La clasificación completa se omite aquí, pero puede consultarse en inglés en la figura 6.5, en la página 101.

Expliquemos brevemente las categorías de sistemas. Hemos visto que los valores de algunas de las cantidades son medibles, en cuyo caso se trataba de cantidades físicas, mientras que otros no, en cuyo caso se trabata de cantidades abstractas. En concordancia, podemos distinguir los sistemas cuyas cantidades son físicas, *sistemas físicos*, del resto, que son *sistemas abstractos*. Si nos concentramos en los sistemas físicos, podremos distinguir aquellos cuyas cantidades realmente existen, de aquellos cuyas cantidades son supuestas. En el primer caso, los sistemas son *sistemas reales*, y en el segundo *sistemas conceptuales*.

En cuanto al número de cantidades y a la estructura, se puede distinguir aquellos sistemas que tienen un número determinado de cantidades y una estructura finita, en cuyo caso son *sistemas acotados*, de aquellos que tienen infinitas cantidades y estructura infinita, que son *sistemas no–acotados*. Podemos deducir que los sistemas reales siempre son acotados, mientras que los sistemas conceptuales pueden ser no–acotados.

Volviendo a analizar las cantidades de un sistema, podemos distinguir dos tipos. En primer lugar, aquellas cantidades del sistema cuyos valores están determinados por el entorno, independientemente del sistema, que son *cantidades independientes*. En segundo lugar, cantidades que pueden depender de otras cantidades del sistema, que son *cantidades dependientes*. Analizando sistemas reales, frecuentemente no se puede averiguar con exactitud cuáles de sus cantidades son dependientes y cuáles independientes debido a la complejidad de las relaciones de un sistema con entorno. En el caso de que esta distinción se conociese, se diría que el sistema es un *sistema controlado*, mientras que de lo contrario sería un *sistema neutral*.

1.2.5 Esta aproximación en términos de la GST

En este texto, analizaremos diversos aspectos de los sistemas autónomos en base a los conceptos y a la terminología de la Teoría de los Sistemas Generales introducida en las secciones precedentes. Algunos de los conceptos serán ampliados en función de lo requerido por el tema desarrollado en cada caso.

Este texto ha sido escrito desde un punto de vista concreto, que podemos definir de acuerdo a los conceptos de sistemas de la GST. En concreto, la exposición se hará entendiendo siempre, como contexto de fondo, un sistema acotado, controlado y secuencial. Expliquemos esto.

Se asume un sistema acotado porque es el caso más común en ingeniería y en sistemas cognitivos. Se asume un sistema controlado porque permite exponer los conceptos claramente y sin ambigüedad. Debe entenderse que, en la realidad, muchas veces no es posible conocer todas las cantidades y relaciones de comportamiento de un sistema, hecho que imposibilita, entre otras cosas, el distinguir entre cantidades dependientes e independientes. En muchos casos comunes en ingeniería, los sistemas se estudian basándose en unas pocas cantidades físicas, que resultan suficientes, y por tanto el sistema puede considerarse controlado. En otros casos, sin embargo, algunos conceptos de los que se han presentado en las secciones precedentes, tales como las transiciones de estados y las relaciones de comportamiento, es preciso entenderlos y estudiarlos asociados a una distribución de probabilidad.

Finalmente, se ha adoptado el punto de vista de un sistema *secuencial*. Primero, porque se considera un caso más general que un sistema *sin memoria*. Segundo, porque se entiende que un requisito básico para que los sistemas puedan tener alta autonomía es que dispongan de memoria. Expliquemos estos conceptos y las diferencias entre sistemas secuenciales y sistemas sin memoria.

Un sistema sin memoria produce respuestas a los valores de sus estímulos en un cierto instante de tiempo, en correspondencia con ellos. Es decir, que si el mismo estímulo se repitiese, el sistema sin memoria repetiría la misma respuesta. En algunas disciplinas estos sistemas se conocen como *combinatorios*, porque su respuesta solo depende de las combinaciones de sus valores de entrada. Sin embargo, la respuesta de los sistemas secuenciales a unos valores concretos de sus cantidades de entrada depende de más factores, como sus valores en el pasado inmediato, como ejemplo más común. Este tipo de comportamiento implica la existencia de algún elemento en el sistema que actúe como memoria, para registrar estos otros factores. Entendemos que un sistema sin memoria es, por tanto, un caso degenerado de un sistema secuencial, en el que la capacidad de almacenamiento del elemento memoria tiende a nula. Por tanto, los análisis que se exponen en este texto referidos a sistemas secuenciales, podrían particularizarse para sistemas sin memoria.

Los sistemas solamente pueden tener comportamiento de alto grado de autonomía si tienen la capacidad de reaccionar adecuadamente a la incertidumbre del entorno. La única posibilidad de que esto ocurra en sistemas de recursos limitados es que su operación esté basada en conocimiento. El tipo y el volumen de conocimiento por un lado, y el modo en que el sistema lo utilice para su operación por otro, determinarán el nivel de autonomía del sistema.

Nos gustaría añadir que algunos aspectos de los sistemas pueden ser analizados desde otros puntos de vista que aportan más información sobre ellos. Tal es el caso del conocimiento. En efecto, el conocimiento puede ser analizado desde el punto de vista *real*, que es el que se desprende de lo anterior, que se refiere a los recursos sobre los que esté implementado: unos recursos limitados de puertas lógicas, consumo de energía, espacio disponible para memoria, etc. Pero los aspectos más relevantes del conocimiento tal vez se deriven de un análisis de su contenido, más que de su soporte, es decir, de la información representada en los citados recursos.

En este segundo caso, podremos tener en cuenta que el significado de la representación contenida en los recursos de memoria del sistema depende también del proceso de interpretación asociado. La misma representación de memoria interpretada de diferentes formas puede tener diferentes significados o ninguno en absoluto. Como ejemplo en relación con el anterior, podemos observar en la vida cotidiana cómo el mismo archivo, una configuración determinada de un conjunto de transistores en un elemento de memoria, puede ser interpretado como un archivo de texto, una imagen, o como algo caótico, dependiendo de la aplicación que se emplee para leerlo. En este caso, podríamos ver que construyendo adecuadamente la representación, ésta podría dar lugar a un conjunto de significados diferentes, o, dicho de otro modo, a múltiples sistemas *conceptuales*.

1.3 Conceptos sobre sistemas autónomos

Intuitivamente, podemos concebir un sistema autónomo como un sistema que es capaz de alcanzar su objetivo en un entorno. Para conseguir ésto, el sistema podrá operar tanto sobre sí mismo, sobre el entorno, o sobre ambos. Pero al mismo tiempo que el sistema evoluciona por su lado, el entorno evoluciona por otro. Su evolución también puede afectar al sistema.

Como resultado de la mutua influencia de sistema sobre entorno y entorno sobre sistema, la forma en que éste se aproxima hacia su objetivo puede cambiar en el tiempo, volviéndose más directa y rápida, más lenta, o eventualmente divergente de él. Los sistemas que son autónomos de forma efectiva tienen la capacidad de mantener la convergencia hacia su objetivo a pesar de la evolución del entorno y de la del sistema en sí.

En esta sección exploraremos los sistemas autónomos para determinar características generales de su operación y de sus propiedades.

1.3.1 Finalidad y directividad

Podemos entender el concepto de *finalidad* de un sistema primeramente como un cierto *objetivo final* al cual un sistema dirige su comportamiento. El término 'objetivo' se refiere a un estado del sistema, del entorno o de ambos.

Un sistema que opera en un entorno, puede evolucionar acercándose a su objetivo o alejándose de él. En el primer caso diremos que presenta un *comportamiento convergente*, y en el segundo, un *comportamiento divergente*.

Llamaremos *directividad* de un sistema a su cualidad de experimentar una evolución no aleatoria. En sistemas autónomos, la directividad representa una tendencia hacia la *finalidad* del sistema.

Sin embargo, el término 'finalidad' tiene más acepciones [vB69, p.77-80]. Tras analizarlas, podemos resumirlas en las siguientes:

1. **Objetivo:** Como se ha mencionado, un estado del par $(sistema, entorno)$, completa o parcialmente definido.
2. **Adecuación:** Aquella aplicación, propósito u objetivo para los que un sistema es adecuado, o que entran dentro de sus capacidades.
3. **Tendencia:** Tendencia del sistema hacia un estado o configuración específica. Expresa la no-aleatoriedad de su comportamiento.
4. **Directividad estructural:** Patrones de comportamiento de un sistema, teniendo en cuenta que derivan de una cierta *organización* del mismo.
5. **Equifinalidad:** La cualidad de un determinado estado de poder ser alcanzado por un sistema partiendo de condiciones y estados iniciales diferentes, y de diferentes formas. Llamaremos *región de equifinalidad* de un sistema al conjunto de todos los estados con la cualidad de equifinalidad respecto a otro.[2]
6. **Directividad de propósito:** Capacidad de un sistema para cambiar su organización, y por tanto su comportamiento, de forma que se establezca, mantenga o mejore su convergencia, mediante consideración simbólica —explícita— de su objetivo, de sí mismo y del entorno.

Consideraremos una semiformalización del concepto de *directividad* como una relación **D**. Indicando el sistema como S, en su entorno E, y el objetivo del sistema, O, podemos escribir la expresión:

$$(S, E)_1 \xrightarrow{\mathbf{D_O}} (S, E)_2 \quad t_1 < t_2$$

Podemos leer que el sistema y el entorno alcanzan el estado $(S, E)_2$ partiendo del $(S, E)_1$, de acuerdo a la directividad del sistema $\mathbf{D_O}$. Si el sistema está siguiendo un *comportamiento convergente*, entenderemos que el par (S, E) estará más cerca de O en t_2 que en t_1, y que ambos estados están dentro de la *región de equifinalidad* de O para el sistema S en el entorno E.

Esta formulación es genérica. A continuación analizaremos la directividad estructural y la de propósito separadamente.

[2] Interesará especialmente el caso en que este último sea un objetivo del sistema.

Directividad estructural

Como hemos mencionado, la directividad estructural de un sistema indica el comportamiento que deriva de una organización específica. Consideremos dos instantes de tiempo, t_1 y t_2, de forma que el primero sea anterior al segundo. Asumamos que el sistema es capaz de compensar las perturbaciones durante el intervalo (t_1, t_2). En estas circunstancias, podemos asumir —a efectos de la convergencia del sistema— que la *región de equifinalidad* en t_1 es equivalente a la del instante t_2. Indiquemos esta región como Γ. Tenemos entonces:

$$(S,E)_1 \xrightarrow{\mathbf{D(S,E)}} (S,E)_2 \quad t_1 < t_2$$

$$(S,E)_1, (S,E)_2 \in \Gamma$$

Esto quiere decir que el comportamiento del sistema, por directividad estructural, llevará al sistema y al entorno del instante t_1, $(S,E)_1$, a un nuevo estado en el instante t_2, $(S,E)_2$. Ambas configuraciones del sistema, que indicaremos por S_1 y S_2, pertenecerían a Γ. Podemos observar que la directividad estructural, representada por **D**, se ha indicado explícitamente dependiente del sistema y del entorno. El objetivo de esta directividad está, por tanto, implícito en S.

Directividad de propósito

La directividad de propósito es la reconfiguración de partes de la organización del sistema a través de procesos que operan con una representación explícita del objetivo del sistema.

Podemos percatarnos de que la directividad de propósito modifica el sistema, S, para que el estado global en el instante t_2, $(S,E)_2$, pertenezca a la región de equifinalidad de O. Ahora el objetivo es explícito en la directividad **D**. El proceso comienza en el instante t_1, en el que la región de equifinalidad es Γ_1:

$$(S,E)_1 \xrightarrow{\mathbf{D(S,E,O)}} (S,E)_2 \quad t_1 < t_2$$

Hay que reparar en el hecho de que los mecanismos de directividad de propósito operan a partir del estado conocido, que es $(S,E)_1$, y sobre una representación de la región de equifinalidad correspondiente, Γ_1. Su operación resulta en $(S,E)_2$. Este estado es asumido como perteneciente a la región de equifinalidad en t_2, que denotaremos como Γ_2. Sin embargo, esto no tiene por qué ser necesariamente cierto, y el sistema podría derivar en comportamiento divergente.

1.3.2 Objetivos

Se puede entender que un *objetivo* es un estado del sistema, del entorno o de ambos a los que el sistema tiende como resultado de su comportamiento.

Como se mencionó previamente, el *estado* del sistema es el valor de todas sus cantidades en un determinado instante de tiempo. El estado del entorno, sin embargo se debe entender *desde el punto de vista del sistema*. Es decir, es una caracterización del entorno expresada en función de las cantidades del acoplamiento sistema-entorno. Por tanto, el estado del entorno se refiere a los valores de estas cantidades. A esto se le llamará el *estado del entorno estricto*.

Pero la noción del entorno con la que el sistema opere no se limita a ésto. El sistema deducirá otros aspectos relativos al entorno en función de estas cantidades, construyéndose representaciones conceptuales de él. El conocimiento del sistema sobre el entorno en un determinado momento estará formado a la vez por las cantidades del acoplamiento, y por todos los aspectos derivados de ellas que el sistema haya podido deducir. A esta noción se le conocerá como *estado del entorno subjetivo*. Salvo indicación expresa en contra, se entenderá esta noción al hablar del *estado del entorno*.

Podemos volver a considerar la noción ideal de objetivo teniendo en cuenta todos estos matices: objetivo como estado del par $(sistema, entorno)$. Los objetivos son *conceptuales*, en tanto a que no existen en la realidad, pero se refieren a ella. En las próximas secciones estudiaremos la influencia de los objetivos en la operación del sistema.

Objetivos y organización

Como se ha mencionado, el comportamiento dirigirá al sistema hacia un objetivo. En sistemas artificiales, el objetivo queda impuesto por el diseñador. En los sistemas biológicos, resulta de la evolución.

Por tanto, el objetivo de un sistema es el factor de donde deriva su directividad. Es decir, las propiedades características de un sistema se corresponden con el objetivo. Un objetivo diferente implicaría propiedades diferentes y, por tanto, un comportamiento diferente. Esto puede resumirse en una relación fundamental de causalidad:

$$\text{objetivo} \longrightarrow \text{organización} \longrightarrow \text{comportamiento}$$

En sistemas complejos esta relación se cumple solo conceptualmente, puesto que pueden coexistir diversos objetivos a la vez, en lugar de uno solo. En este caso, un objetivo determinaría una parte de las propiedades del sistema, y a su vez éstas determinarían aspectos parciales del comportamiento.

Estructura de objetivos

Podemos asumir que los objetivos de un sistema forman un conjunto heterogéneo; que cada objetivo puede ser diferente de los demás. Estas diferencias pueden ser de diversa naturaleza, pero podríamos clasificarlas en dos:

Alcance temporal: La duración del período necesario para que el sistema alcance el objetivo.

Nivel de abstracción: Los objetivos que se refieren a cantidades físicas del sistema son de bajo nivel de abstracción. Aquellos referidos a cantidades abstractas son de alto nivel. Mayor dependencia de cantidades abstractas implica mayor nivel de abstracción.

El nivel de abstracción y el alcance temporal de un objetivo no son aspectos independientes. De hecho, normalmente un mayor nivel de abstracción irá unido a un mayor alcance temporal.

Sin pérdida de generalidad, podemos asumir que los objetivos de un sistema se organizan de acuerdo a una estructura de dependencia jerárquica, a la que llamaremos *estructura de objetivos*. Los objetivos de bajo nivel de abstracción y corto alcance temporal contribuirán a realizar objetivos de mayor alcance y abstracción. Para referirnos a unos objetivos respecto a los otros, los distinguiremos por *de mayor nivel* o *más altos* por un lado y de *menor nivel* o *más bajos* por el otro.

Por el extremo de objetivos de menor nivel de abstracción y alcance, la estructura de objetivos estaría formada por *objetivos locales*. En el extremo de mayor abstracción y alcance se compondría de *objetivos raíz* o *generadores*. Entre ambos extremos existirían los *objetivos intermedios*.

La jerarquía de objetivos se puede ver metafóricamente como una cascada en cuya cumbre se encuentran los objetivos raíz, que se *descomponen* en objetivos intermedios, y éstos a su vez en otros, hasta alcanzar la base, formada por objetivos locales.

En resumen, un sistema autónomo tiende a realizar sus objetivos raíz a través de otros de menor nivel, que son o bien más simples, o bien de más corto alcance temporal. El comportamiento del sistema tiende a alcanzar todos los objetivos de la estructura progresivamente, siguiendo su jerarquía de dependencias. Por tanto, se puede decir que la estructura de objetivos realmente define la tendencia en la evolución del sistema: su directividad.

Como hemos visto, los objetivos pueden ser diferentes en cuanto a alcance temporal y niveles de abstracción. Esto implica que unos objetivos se alcanzan para realizar otros de mayor nivel, y que objetivos nuevos pueden aparecer. En definitiva, que la estructura de objetivos presenta una cierta dinámica. Esto es un factor para la autonomía del sistema, si consideramos algunos aspectos:

- Mientras los objetivos raíz permanezcan inalterados, el resto se pueden crear, eliminar o modificar dinámicamente.

- De esta forma, la directividad del sistema se puede adaptar a los cambios en el escenario de operación para preservar la convergencia a los objetivos raíz.

- El hecho de tener los objetivos raíz descompuestos en otros de menor alcance temporal y nivel de abstracción incrementa la tolerancia a perturbaciones. Una situación puntual de divergencia en objetivos inferiores, derivada de una perturbación, afectaría solamente a partes aisladas de la estructura. Las partes afectadas serían más extensas cuanto más alto el objetivo.

- Por tanto, sería posible modificar solamente partes de la estructura dejando el resto inalteradas.
- Disponer de múltiples objetivos permite operación dedicada separadamente que puede ser ejecutada en paralelo.

Categorías de objetivos

Una vez matizados estos aspectos en cuanto a los objetivos de un sistema, podemos aplicarlos a la relación fundamental de causalidad ya mencionada obteniendo las siguientes relaciones:

$$\begin{array}{rcl} \text{objetivos raíz} \rightarrow & \text{estructura real} & \rightarrow \text{comportamiento permanente} \\ \text{objetivos intermedios} \rightarrow & \text{estructura hipotética} & \rightarrow \text{comp. relativamente permanente} \\ \text{objetivos locales} \rightarrow & \text{programa} & \rightarrow \text{comportamiento transitorio} \end{array}$$

Es decir, que los objetivos raíz se corresponden con las propiedades que motivan el comportamiento real del sistema, los intermedios con la estructura hipotética y el comportamiento relativamente permanente, y finalmente los locales con el programa y el comportamiento transitorio.

Dinámica de objetivos

Como se ha mencionado anteriormente, la estructura de objetivos sigue una cierta dinámica, que resulta de la realizar objetivos locales e intermedios, y de crear otros nuevos. La dinámica de un objetivo concreto seguirá una secuencia típica de fases:

$$\text{generación} \rightarrow \text{activación} \rightarrow \text{actividad} \rightarrow \left[\begin{array}{l} \text{desactivación} \\ \text{conclusión} \end{array} \right.$$

Por *generación* se entenderá el proceso necesario en el sistema para crear un objetivo; típicamente un proceso de inferencia o de planificación. La *activación* se refiere a la preparación previa del sistema para seguir el objetivo, por ejemplo: asignación de recursos y tiempos de cómputo. La *actividad* de un objetivo es el período durante el cual el sistema muestra un comportamiento convergente a él. La *desactivación* consiste en la eliminación de un objetivo de la estructura de objetivos del sistema, de forma que cese su actividad antes de ser alcanzado. Por último, la *conclusión* es la realización del objetivo —su fin natural—.

1.3.3 Organización

En esta sección trataremos de analizar el sistema autónomo en términos de su organización, relacionando ésta con los objetivos y con el comportamiento del sistema, para construir una visión unificada y global.

El sistema en términos de objetivos, organización y comportamiento

Adoptemos una perspectiva global del sistema autónomo. Como hemos mencionado, la *organización* de un sistema en un momento dado se corresponde con su *estructura de objetivos*. Esto quiere decir que las *propiedades*[3] del sistema hacen que éste evolucione hacia los objetivos. A esta evolución le llamamos *comportamiento*.

Hemos visto que existen diferentes tipos de *objetivos*, *organización* y *comportamiento*. Los *objetivos raíz* se corresponden a un conjunto de propiedades que forman la *estructura real* del sistema, y que es la causa de su *comportamiento permanente*. Los *objetivos intermedios* se corresponden con otro conjunto de propiedades que forma la *estructura hipotética* y causa el *comportamiento relativamente permanente*.

Debemos concebir la *estructura real* del sistema como una predisposición o adecuación básica del sistema —segundo sentido de *finalidad*—.

Miremos ahora el sistema desde otra perspectiva, volviendo a la formulación de la GST. La organización del sistema se puede explicar en base a dos conceptos: la *estructura-UC* y la *estructura-ST*. El primero explica los *elementos* que forman el sistema y cómo están relacionados los unos con los otros. El segundo explica las posibles configuraciones y valores que pueden adoptar.

En estos términos, el *comportamiento permanente* del sistema, derivaría de la *estructura-UC*, dentro de las posibilidades derivadas de la *estructura-ST*. Como se ha mencionado arriba, el comportamiento permanente sigue una tendencia básica del sistema hacia los *objetivos raíz*.

Sin embargo, eventualmente los *objetivos intermedios* y *locales* pueden cambiar, causando que algunos de los elementos del sistema se modifiquen o sean sustituidos por otros. Esto se refleja en el conjunto de estados potencialmente alcanzables por el sistema. Los elementos y acoplamientos susceptibles de sustitución o modificación constituyen la *parte variable* de la *estructura-UC*, y los estados potenciales derivados, la parte variable de la *estructura-ST*. Derivan en los comportamientos *relativamente permanente* y *transitorio*.

Como se avanzó arriba, la dinámica de la *estructura de objetivos* y por tanto, la de la *organización*, representan la adaptatividad del sistema. Esto deriva de la posibilidad del sistema para modificar sus propiedades adaptándolo a las perturbaciones y a la evolución del escenario de operación.

Funciones

Entenderemos una *función* como una sucesión de estados asociados a un objetivo específico. Es decir, un conjunto de estados y transiciones que contribuye a realizar un objetivo. La función es, por tanto, un *subprograma*, en el sentido de la terminología GST [Kli69, p.45], [Kli80, p.80]. Siguiendo la secuencia de estados y transiciones del subprograma, el sistema avanzará hacia el objetivo.

[3] Debe recordarse que se llama *organización* del sistema al conjunto de sus *propiedades*.

Se puede asumir que, en el caso más general, un objetivo dado puede alcanzarse de múltiples maneras, a través de diversas funciones. También puede componiendo varias funciones, en lugar de una sola. Dado un objetivo, llamaremos *descomposición funcional* al proceso por el cual se le asigna una función concreta o un conjunto de funciones.

Desde un punto de vista cognitivo, la relevancia de una función es su *algoritmo*. Un *algoritmo* especifica un modo particular de alcanzar el objetivo correspondiente. Los algoritmos pueden guardarse representados en la memoria del sistema. En este caso se dice que estas representaciones constituyen funciones *en forma conceptual*. Cuando el sistema está adaptado a un algoritmo específico, se dice que la función está *corporizada*.

Independientemente de estar en forma conceptual o coporizadas, podemos distinguir tres aspectos de las funciones, que llamaremos *aferente*, *eferente* y *deliberativo*. Se corresponden aproximadamente con los conceptos usuales de *entrada*, *salida* y *procesamiento* respectivamente.

La evolución de las cantidades aferentes representan la entrada a la función, intuitivamente su parte *perceptiva*. Análogamente, las cantidades eferentes representan la salida o resultado de la función, su parte de *acción*. Conviene matizar que el término 'acción', aunque se utiliza comúnmente y es intuitivo, es impreciso. La parte eferente de una función no tiene por qué conllevar una acción física. Debe entenderse, en el caso general, como el proceso de cambiar el estado de un cierto número de cantidades que, a veces pero no necesariamente, pueden ser cantidades físicas. Por último, la evolución de las cantidades deliberativas representa el procesamiento o cómputo asociado a la función, intuitivamente, el *pensamiento*.

Algoritmos y funciones corporizadas

Hemos introducido las nociones de *algoritmo* y *función corporizada* como una sucesión de estados y como su implementación con cantidades reales. No se puede asumir en el caso general que una función corporizada se corresponda directamente con su algoritmo. Está sujeta a restricciones derivadas del escenario de operación y de los recursos del sistema. En concreto, podemos considerar los siguientes puntos:

- Una función corporizada debe estar basada en un conjunto específico de cantidades del sistema. Sin embargo, un algoritmo puede estar formulado idealmente, sin hacer referencia a cantidades concretas. Esto significa que los valores de algunas de las cantidades consideradas pueden no estar especificados por el algoritmo. Por tanto, hacer corresponder un algoritmo con un conjunto específico de cantidades implica resolver la indeterminación. En otras palabras, el proceso de corporización de un algoritmo debe asignar valores arbitrariamente a aquellas cantidades indeterminadas. Esto puede derivar en comportamiento emergente.

- Un algoritmo puede o puede no especificar aspectos temporales. En el primer caso, las prestaciones de los recursos del sistema deben ser suficientes para asegurar el cumplimiento de la especificación. De lo contrario, el algoritmo real de la función corporizada no se corresponderá con el original. En el caso de que los aspectos temporales no estén especificados, esto puede deberse a dos causas.

Primero, que el tiempo sea irrelevante para el algoritmo y por tanto para el objetivo. Esto supone un grado de libertad a la hora de corporizarlo. Segundo, que el tiempo sea relevante, pero no esté especificado. Esto requeriría resolver esta indeterminación durante el proceso de corporización.[4]

- Los algoritmos conocidos por el sistema pueden provenir de múltiples fuentes, por ejemplo: bases de datos, aprendizaje o procesos de resolución de problemas. Debido a ésto, puede ocurrir que un algoritmo no esté perfectamente adecuado al sistema o al escenario concreto en el que se va a aplicar. Esto puede tener diversos efectos. El ejemplo más inmediato sería el caso en que los estados del algoritmo no pudieran alcanzarse, por estar fuera del rango de las cantidades reales del sistema (insuficiente potencia, resistencia mecánica, etc.)

- El conjunto de recursos necesarios para corporizar un algoritmo concreto puede ser también necesario para otros algoritmos. Esto implicaría la necesidad de compartir recursos o priorización. Una gestión de recursos ineficaz causará típicamente comportamiento emergente y fallos en el cumplimiento de las especificaciones temporales.

Estructura funcional

Como se ha mencionado anteriormente, las funciones que lleva a cabo un sistema están asociadas a sus objetivos. Por tanto, los objetivos que forman la estructura de objetivos en cada instante, tendrán un conjunto de funciones asociadas, también siguiendo una estructura, que llamaremos *estructura funcional* del sistema. En un caso ideal, la estructura funcional podría corresponderse directamente con la de objetivos. En la realidad, los recursos sobre los que las funciones están corporizadas, y las circunstancias impuestas por el escenario de operación, pueden hacerlos diferir, como se deduce de las consideraciones efectuadas arriba.

Los objetivos se descomponen en una o más funciones que los realizan, por lo que la correspondencia entre las estructuras funcional y de objetivos no será en general directa, *uno a uno*. Además, como mencionábamos antes, las funciones están corporizadas en recursos reales del sistema, los que pueden introducir nuevas dependencias y restricciones derivadas del substrato físico. En conclusión, aunque existirá una semejanza entre las topologías y las relaciones de dependencia de la estructura de objetivos y la de funciones, no serán iguales.

Como se mencionó, la adaptatividad del sistema depende de que la estructura de objetivos se refleje adecuadamente en la funcional, como resultado de un proceso de descomposición de los objetivos en funciones que los realicen, que se llamará, como se ha mencionado, *descomposición funcional*. Este proceso es, por tanto, uno de los factores para la autonomía del sistema.

[4]Cualquier indeterminación en un algoritmo, sea en cuanto a cantidades o a tiempo o de otra índole, puede causar comportamiento emergente.

Analicemos la operación del sistema en conjunto, desde este punto de vista, para determinar de qué forma tiene lugar la adaptatividad en el sistema. Como se ha mencionado, una estructura de objetivos puede ser realizada por múltiples estructuras funcionales. En otras palabras, que diferentes conjuntos de sucesiones de estados pueden llevar al mismo estado final. De esto se deduce que si el sistema falla en alcanzar un objetivo específico, puede adoptar dos alternativas posibles para mantener la convergencia:

- Primero, redefinir la estructura de objetivos —en la medida en que fuera necesario— para preservar la finalidad del sistema.
- Segundo, sustituir el algoritmo del objetivo inalcanzado —sin modificar éste último— llevando a cabo un nuevo proceso de descomposición funcional.

La reconfiguración de la estructura de objetivos, el proceso de descomposición funcional y el proceso de gestión de los anteriores constituyen los principales mecanismos de directividad del sistema.

Comportamiento anómalo

Volviendo a una perspectiva global, se puede decir que el *comportamiento* del sistema es el resultado de sus funciones corporizadas operando en un entorno. Desarrollemos esta idea básica teniendo en cuenta la noción de función como *subprograma*.

Una función es una sucesión particular de estados con un conjunto de transiciones asociadas. Evidentemente, estos estados deben contarse dentro de los posibles del sistema,[5] para que una función pueda ser corporizada.

Por otro lado, cuando un sistema está ejecutando una función corporizada, sigue la secuencia de estados *automáticamente*, iniciando la siguiente transición una vez se ha alcanzado cada estado. Esta dinámica está especificada en el algoritmo. Idealmente, el algoritmo está corporizado de tal manera que el sistema solamente sigue los estados y transiciones especificados.

Este no es el caso real, por las consideraciones expuestas anteriormente. Puede ocurrir que la función corporizada no pueda alcanzar exactamente el estado especificado en su algoritmo. En este caso, el sistema puede abandonar definitivamente la convergencia o eventualmente recuperar la secuencia de estados especificada. También puede alcanzar estados —y sucesiones de éstos— imposibles de abandonar.[6] A este tipo de comportamiento, en el cual el sistema se separa de la sucesión de estados dada por el algortimo, se le llamará *comportamiento anómalo*. *>sec. 7.4.5, p. 126 y fig. 7.2, p.7.2.*

Podemos resumir las principales causas para el comportamiento anómalo, expuestas previamente, en dos categorías: las debidas a indeterminación en el algoritmo y las debidas a restricciones impuestas por el substrato.

[5] El conjunto de los estados posibles de un sistema junto con las transiciones se llama *programa* o *programa completo* [Kli80, p.80], *program*, *complete program* [Kli69, p.45].

[6] Un ejemplo serían los *ciclos límite* en los sistemas.

El conocimiento puede ayudar a eliminar o reducir el comportamiento anómalo de un sistema. Mejor conocimiento se traduce en mejores algoritmos: mayor precisión y transiciones mejor planificadas por un lado, y una mejor modelización del substrato del propio sistema, por otro. Mejor conocimiento también implica mejores procesos de descomposición funcional: mejor selección de algoritmos, y mejor correspondencia entre algoritmo y función corporizada.

Modelo de nodos y flujos

El *modelo de nodos y flujos* es una herramienta para modelizar la estructura funcional de los sistemas autónomos, diseñada con la intención de mostrar explícitamente la estructura de dependencias funcionales, así como los aspectos cognitivos de las funciones (aferente, eferente, deliberativo).

Podemos distinguir dos tipos de dependencia dentro de la estructura funcional. En primer lugar, consideremos el caso de un objetivo descompuesto en múltiples funciones, cada una definida por su algoritmo. Necesariamente, estos algoritmos están relacionados, al menos, por la cualidad de contribuir al mismo objetivo. De hecho, los algoritmos podrán compartir información. Este caso es un ejemplo del primer caso de dependencia: *dependencia cognitiva*.

Por otro lado, como hemos visto, las funciones corporizadas difieren de sus respectivos algoritmos en una serie de restricciones derivadas del substrato sobre el que están implementadas. Es decir, que las capacidades y la operación reales de las funciones dependen del sustrato en el que están corporizadas. Estas restricciones son en realidad *relaciones de comportamiento* entre sus cantidades: interdependencia. Se llamará *dependencia estructural* o *dependencia de substrato*.

El modelo de nodos y flujos muestra ambos tipos a través de los conceptos de *nodo* y *flujo*. Ambos conceptos modelizan lo mismo: una función. Sin embargo, cada uno permite enfatizar diferentes aspectos de la misma función. Los emplearemos conjuntamente.

Nodos: Un *nodo* representa una función corporizada. También emplearemos el término *unidad funcional* indistintamente, aunque el primero generalmente hará referencia a la función entendida dentro de la estructura funcional, y el segundo, como función aislada.[7]

Como sabemos, la estructura funcional es un conjunto de funciones mutuamente dependientes y relacionadas. Por tanto, una topología de nodos. Las dependencias entre ellos pueden ser de cualquier tipo, por ejemplo, jerárquicas en el sentido *cliente–servidor* o *maestro–esclavo*, o dependencias de substrato, como en el caso de funciones que compartiesen recursos.

[7] No olvidemos que en último término una función corporizada equivale a unas relaciones de comportamiento concretas entre un conjunto específico de cantidades reales del sistema, ajustadas a las posibilidades y finalidad de éste.

Un nodo es una estructura compuesta por cuatro *elementos* o *componentes*: *aferente*, *eferente*, *deliberativo/central* e *integrador*. *>fig. 7.3, p.130.* Los tres primeros se refieren, respectivamente, a las partes de la función, de acuerdo con lo mencionado previamente. El cuarto elemento representa los aspectos comunes, estructurales de la unidad funcional. Desde el punto de vista de estados y transiciones, el elemento integrador comprende un conjunto de subprogramas que proporcionan estados auxiliares y los acoplamientos del resto de la estructura funcional con los demás elementos del nodo. Desde un punto de vista cognitivo, proporciona a los demás elementos del nodo mecanismos de sincronización, comunicaciones, gestión de recursos y todos los demás aspectos estructurales.

Debemos tener en cuenta algunas consideraciones respecto a la modelización con nodos. La estructura de nodo que acabamos de describir representa una generalización y una conceptualización de la noción de función. Por tanto, las funciones concretas que se puedan identificar en un sistema pueden no tener alguno de los tres componentes. Por ejemplo, funciones de un sistema concentradas en la percepción, con toda probabilidad tendrán unos elementos central y aferente muy reducidos o inexistentes. En este caso, se diría que los elementos están *degenerados*.

También hay que tener en cuenta que modelizar un sistema con nodos es arbitrario, a juicio del investigador. Un mismo sistema podría modelizarse de diversas formas, siguiendo diferentes criterios o situándose a diferentes niveles de abstracción. Así, podríamos considerar el sistema como un único nodo, o dividir la operación de este en multitud de nodos elementales. También podríamos considerar unas funciones como subfunciones de otras, dando lugar a *nodos anidados* —nodos dentro de otros—.

Flujos: Como se ha dicho arriba, los *flujos* sirven para modelar funciones resaltando aspectos diferentes a los nodos. Mientras éstos resaltan aspectos cognitivos —explican la función de acuerdo a cuatro tareas cognitivamente distintas— un *flujo* distingue los aspectos operacionales de los de implementación. También consta de cuatro elementos *>fig. 7.4, p.133*:

- Interfaz de entrada.
- Interfaz de salida.
- Unidad de ejecución.
- Definición funcional.

El funcionamiento normal del flujo consiste en (1) la aparición de valores en las cantidades que forman el *interfaz de entrada*, que es el acoplamiento con el entorno o con el resto del sistema. (2) La *unidad de ejecución* transforma los valores de las cantidades del interfaz de entrada. Constituye el proceso asociado a la función; en términos computacionales: buffers, operadores, registros, etc. (3) Finalmente, el resultado del proceso pasa a las cantidades del *interfaz de salida*, que están acopladas al entorno o al resto del sistema.[8]

La *definición funcional* es una especificación de la función corporizada que realizará el flujo. Por tanto, define cómo son todos los elementos del flujo, en concreto:

- Conjunto de recursos asociados al flujo.
- Estructura del flujo (interfaces, unidad de ejecución, estados y transiciones posibles).
- Nivel de resolución en el que operará el flujo.
- Relaciones entre las cantidades asociadas (corporización de los algoritmos).

El nivel de resolución y las especificaciones de los interfaces vendrán dados —en gran medida— o limitados por la estructura funcional, ya que son los acoplamientos entre el flujo y el resto de flujos y el entorno. El conjunto de algoritmos y recursos dependerá de la disponibilidad del sistema en cada instante, y en el objetivo asociado a la función del flujo.

Modelo de nodos y flujos: Se trata de una combinación de nodos y flujos, aunque, como se ha mencionado previamente, éstos se podrían utilizar por separado. Recordaremos que los *nodos*, al distinguir entre elementos funcionales aferentes, eferentes, centrales e integradores hacen explícitas las relaciones cognitivas entre componentes. Los *flujos* hacen explícitas las dependencias estructurales o computacionales, al separar interfaces de unidades de ejecución y definiciones funcionales.

En esencia, el *modelo de nodos y flujos* consiste en considerar cada elemento de un nodo como si fuera un flujo. Es decir, que el elemento aferente se modeliza con un flujo dedicado a procesos aferentes, el eferente con un flujo especializado en procesos eferentes, y análogamente con los otros dos elementos del nodo. *>fig. 7.5, p.134.*

Es interesante analizar las interacciones entre los elementos en el modelo de nodos y flujos. *>fig. 7.6, p.136.* Desde un punto de vista cognitivo, el nodo *interpreta* su entorno con el elemento aferente, procesa la información con el central y ejecuta sus acciones —físicas o no— con el elemento eferente. El elemento integrador coordina a los otros tres elementos de forma que su operación sea consistente con el objetivo del nodo —de la función a la que representa—.

Globalmente, por tanto, los elementos aferente y eferente se comportan como si fuesen "interfaces de entrada" y "de salida" del nodo. Sin embargo, no se limitan a ser interfaces —en el sentido de los interfaces de la noción de *flujo*— sino que son en realidad *intérpretes;* los procesos que llevan a cabo pueden llegar a ser muy complejos. Interpretan el estado del entorno *en relación* al objetivo del nodo,

[8]Se puede observar la analogía entre nodos y flujos. Los interfaces de entrada y salida realizan funciones aferente y eferentes elementales: degeneradas.

o bien trasladan un resultado expresado en estos términos a los términos del acoplamiento con el entorno.

1.3.4 Autonomía

Podemos retornar a las acepciones del término *autonomía* mencionadas previamente y añadir *finalidad* explícitamente para clarificar la exposición:

1. Independencia del entorno.
2. Cohesión del sistema.
3. Finalidad (directividad hacia objetivos).

Analicemos el proceso por el cual el sistema puede llegar a perder la cohesión y a abandonar el comportamiento convergente.

La incertidumbre del entorno afectará al sistema en forma de *perturbaciones*. El *programa* del sistema tiene cierta capacidad de compensar estas perturbaciones. Llamaremos *prestaciones* a esta capacidad. Las prestaciones, efectivamente, equivalen a la eficacia del comportamiento transitorio del sistema. Sin embargo, las prestaciones pueden llegar a ser insuficientes para hacer frente a algunas perturbaciones. En este caso se produce lo que llamaremos *fallo de programa*. *>sec. 7.5.2, p.140 y fig. 7.8, p.140.*

Las consecuencias de un fallo de programa pueden afectar a la *estructura hipotética* del sistema. A este nivel, los mecanismos de directividad pueden activarse para tratar de reconfigurar el sistema corrigiendo su comportamiento. Esto puede consistir en modificar algoritmos o en reconfigurar regiones más o menos extensas de la estructura funcional. Llamaremos a esta cualidad del sistema *adaptatividad*. Podemos deducir de las discusiones previas en cuanto a objetivos y directividad, que la adaptatividad de un sistema puede ser *estructural*, en el caso de obedecer a una función de la actual estructura funcional, o bien *de propósito*, en el caso en que se desarrolle dinámicamente (podemos asumir que este segundo caso implica operación simbólica). Puede darse el caso de que la adaptatividad del sistema no logre compensar el fallo de programa. Llamaremos a esta situación *fallo estructural*.

El fallo estructural se puede propagar a la estructura real del sistema. Puede romperse —total o parcialmente— la *cohesión* del sistema. Esto se conocerá como *degradación* del sistema. La forma concreta en que se dé esta situación dependerá del caso.

Prestaciones

Podemos ver que la autonomía de un sistema equivale, en cierto sentido, a sus prestaciones y adaptatividad. Las prestaciones representan la capacidad del sistema de mantener un comportamiento transitorio convergente frente a algunos tipos de perturbaciones. Podemos deducir de secciones anteriores que las prestaciones de una función dependen fundamentalmente de tres aspectos:

- Precisión de las especificaciones de la secuencia de estados.

- Viabilidad de las transiciones de estados especificadas.
- Completitud de la especificación.

Como se mencionó, la especificación de una función puede contener indeterminaciones o definir transiciones demasiado exigentes para el sistema. Como se vio, esto puede llevar a comportamiento anómalo. La tendencia al comportamiento anómalo hace a la función más vulnerable ante perturbaciones.

Adaptatividad

Podemos observar que la adaptatividad de un sistema depende esencialmente de dos tipos de procesos: *reconfiguración de objetivos* y *descomposición funcional*. Como hemos mencionado, la descomposición funcional de un objetivo concreto no es un proceso unívoco, por lo que diferentes descomposiciones del mismo objetivo pueden ser igualmente posibles. Las fases generales de un proceso de descomposición funcional son las siguientes:

- [Generación de algoritmo.]
- Selección de algoritmo.
- Corporización.

La fase de 'generación de algoritmo' está representada entre corchetes para indicar que puede tener lugar independientemente de las otras dos.[9] Podemos distinguir cuatro niveles de descomposición funcional:

1. Mantener el algoritmo actual y re-corporizarlo con el fin de adaptarlo mejor al escenario de operación actual.
2. Seleccionar un nuevo algoritmo —de entre las alternativas dadas por el conocimiento del sistema— y corporizarlo.
3. Generar un nuevo algoritmo dinámicamente y corporizarlo.
4. Finalmente, cualquiera de estas alternativas puede resultar inviable o insuficiente, lo que haría necesario redefinir parte de la estructura funcional. Cambios mayores incluso podrían requerir modificaciones de la estructura de objetivos. Este fenómeno, por el cual se adaptan niveles de la organización progresivamente más estructurales lo llamaremos *propagación de adaptatividad*.

La *reconfiguración de objetivos* también sigue varias fases:

- [Generación de objetivos.]

[9] Es decir, que pudieran existir en el sistema procesos especializados de generación de algoritmos, que guardasen sus resultados en la memoria del sistema en previsión de eventualidades futuras, y que tuviesen lugar sin necesidad de responder a un proceso de descomposición funcional en curso.

- Selección de objetivos.
- Descomposición funcional.

Como en el caso de 'generación de algoritmos', la generación de objetivos puede ser independiente de las demás fases, y también puede ser activada por las otras dos.

Podemos concluir de esta visión de la autonomía de un sistema, que ésta queda determinada por: prestaciones —entendidas como algoritmos eficientes y corporizaciones eficientes—, y capacidades de generación de objetivos y algoritmos. Observamos que estos aspectos se refieren a dos facetas del sistema: el sustrato en el que está implementado, y el conocimiento y procesos abstractos con que opera.

Principios de autonomía

Podemos observar que hay una pequeña colección de factores que posibilitan altos niveles de autonomía en un sistema. Los llamaremos *principios de autonomía*, para recalcar que se trata de principios de diseño para sistemas artificiales —aunque algunos de ellos se pueden observar claramente en ejemplos biológicos—. Se pueden expresar como sigue:

Mínima estructura. Según la Teoría General de Sistemas, la *organización* de un sistema puede dividirse en dos partes: *estructura* y *programa*. El *principio de mínima estructura* indica que altos niveles de autonomía requieren minimizar la estructura del sistema, lo que implica maximizar su programa. Esto equivale en primer lugar a maximizar las prestaciones. En cuanto a la estructura, equivale a minimizar la estructura real frente a la hipotética. Estas reglas posibilitan máxima adaptatividad.

Encapsulación. Este principio equivale a dos aspectos. Primero, a minimizar los acoplamientos entre los elementos del sistema. Segundo, a construir interfaces que unifiquen y encapsulen conjuntos de elementos heterogéneos.

La encapsulación contribuye a la autonomía de diferentes maneras. En primer lugar, la minimización de acoplamientos es una forma de reducir la estructura del sistema. En segundo lugar, la encapsulación favorece la reconfigurabilidad —modularidad— en sus dos aspectos fundamentales: *separabilidad* y *recombinabilidad* de elementos.[10] Por último, la encapsulación facilita la (auto-)modelización del sistema y por tanto, todos los procesos asociados: autoaprendizaje, representación de conocimiento, simulación, corporización de algoritmos, asignación de recursos, descomposición funcional, etc.

Homogeneidad. El principio de homogeneidad se refiere a la similaridad entre los elementos del sistema.

[10]Para una introducción a la modularidad ver [SP05]. Estos dos términos son traducción de *recombinability* y *separability*.

Este es un factor que contribuye principalmente a la adaptatividad, incrementando la recombinabilidad de los elementos del sistema. Un sistema con elementos homogéneos podría eventualmente dedicar cualquiera de ellos al proceso más exigente en cuanto a recursos, y sustituir un elemento defectuoso por cualquier otro. Elementos heterogéneos y especializados producirían el efecto contrario, e incrementarían las restricciones estructurales del sistema. Por otra parte, los elementos homogéneos son más fáciles de modelizar, produciendo efectos análogos en cuanto a la (auto-)modelización que el principio de encapsulación.

En sistemas reales, la homogeneidad total no es posible salvo en casos excepcionales.[11] En estos casos, el grado de homogeneidad puede incrementarse mediante la adición de elementos intermedios llamados *interfaces*. Estos elementos permiten acoplar elementos heterogéneos.

Isotropía del conocimiento. Este principio se refiere a la cualidad del conocimiento —y de su representación— de ofrecer significados coherentes ante interpretaciones desde contextos de operación dispares.

Podemos observar que el conocimiento es generado en los sistemas desde un punto de vista asociado a un escenario de operación concreto. Es decir, que el conocimiento es generado por una estructura funcional concreta y asociada a unos objetivos concretos, que influyen en él. Llamaremos *especificidad* a esta influencia.

La isotropía perfecta se daría cuando el conocimiento del sistema pudiese ser explotado independientemente de su especificidad. De modo que el mismo conocimiento pudiese tener sentido en diferentes escenarios de operación, aumentando la utilidad potencial de éste.[12]

Escala y escalabilidad. Los aspectos previos tratan aspectos constitutivos de los sistemas. El *principio de escala y escalabilidad* se refiere a la capacidad del sistema para *crecer*.

En diferentes puntos a lo largo del texto, se ha mencionado la importancia del conocimiento, la adaptatividad y otros aspectos. La disponibilidad de recursos es un factor crucial para todos ellos. Mayores recursos implican potencialmente mayores posibilidades de reconfiguración y mayores programas —y mayores prestaciones—. Pueden llegar a ser un factor en contra de la autonomía si implican un mayor número de cantidades independientes o una mayor estructura.

El *principio de escala* se refiere a la realización de los demás principios a través del crecimiento del sistema. Se utiliza el término 'crecimiento' para recalcar que el proceso es el resultado tanto de incrementar recursos como de integrarlos de acuerdo a la directividad del sistema.

[11] El capítulo 14 analiza un sistema tolerante a fallos basado en componentes homogéneos. En él se puede ver cómo los mecanismos de adaptatividad del sistema se basan en esta cualidad.

[12] Obsérvese que puede llegar a ser muy improbable que una situación idéntica se dé en un entorno no controlado más de una vez. Por tanto, si el conocimiento obtenido en unas circunstancias fuera de utilidad en otras, el sistema dispondría de más recursos potenciales.

Chapter 2

Percepción

2.1 Revisión de los estudios sobre percepción

En esta sección realizaremos un recorrido breve por los estudios que adoptan una perspectiva global sobre la percepción. Existen otros estudios que se concentran en aspectos concretos del fenómeno tales como los sentidos, la formación de conceptos o la neurofisiología de la percepción. La parte inglesa >*cap. 8, p.153* incluye un resumen de estas líneas de investigación.

Como decíamos, resumiremos aquí los estudios que se tratan la percepción globalmente, es decir, en relación al sistema observador en el que tienen lugar:

- La función de la percepción en el sistema. Relevancia para la autonomía, comportamiento guiado por objetivos, prestaciones, etc.
- Causas de los fenómenos perceptivos: planteamientos relativos al sistema para explicar percepciones ilusorias, alucinaciones y otros fenómenos similares, y también aspectos generales como la percepción del movimiento, de volúmenes, etc.
- Ciencia (neuro-)cognitiva: relación de los conceptos, su formación y reconocimiento con el substrato neuronal.

La mayoría de estos estudios se basan en el estudio de sistemas biológicos, especialmente del ser humano. Se han llevado a cabo principalmente en psicología. A continuación describiremos las tendencias fundamentales, distinguiendo dos categorías principales: la percepción *directa* y la *indirecta*.

1. Percepción directa. También llamada *aproximación ecológica*. Propuesta por J. J. Gibson en dos trabajos principales [Gib66] [Gib87].

Frente a otras tendencias que asumen que la percepción es un proceso de inferencia por el cual se busca una explicación a las lecturas de los sentidos, esta teoría propone que la percepción se lleva a cabo directamente. Es decir, que no hay un proceso de inferencia, y que las lecturas sensoriales en sí contienen un significado coherente para el sistema.

En concreto, de acuerdo a esta teoría, el sistema percibe *affordances*. Se podría traducir este término aproximadamente por *utilidad*. Es decir, que el sistema percibe directamente, a través de sus sentidos, la utilidad —para sí— que se deriva del estado actual del entorno. Ejemplos de *utilidad* pueden ser: 'soporte', es decir, superficies que pudieran constituir un soporte para el sistema, y 'nutriente', sustancias que pudieran servir de alimento.

2. Percepción indirecta o mediada. En algunos contextos se conoce como aproximación *constructivista* o *computacional*. Asume que la percepción es un proceso de inferencia, que depende no sólo de la lectura de los sentidos, sino también de aspectos propios del sistema como su conocimiento, experiencia previa, emociones, etc. Por tanto, lo que finalmente percibe el sistema estaría *mediado* por sí mismo.

Esta línea de teorías es la más prolija. Se puede destacar las siguientes aproximaciones concretas:

Aproximación de Helmholtz. Se le atribuye la primera formulación de la percepción como proceso de inferencia, en su obra clásica [vH05]. De acuerdo a su teoría, el observador *añade* información a la obtenida de los sentidos. Esta información añadida ayuda al sistema a inferir el estado del entorno como *la explicación más probable*.

Umwelt—Jakob von Uexküll. Desde otra perspectiva, el concepto de *umwelt* [vU82] hace referencia al entorno que realmente percibe el observador. Este entorno sería una versión de la realidad distorsionada por el sistema en función de sus deseos, estados, miedos, etc. En ocasiones el término *umwelt* se traduce por *entorno subjetivo*.

Gestalt. La escuela de psicología Gestalt surge en la primera mitad del siglo XX, tratando de explicar la percepción y otros fenómenos desde un punto de vista sistémico. En concreto, la percepción según esta teoría consiste en un proceso de inferencia para deducir el estado del entorno que *mejor* explicaría la lectura de los sentidos.

Percibir consistiría entonces en analizar la estructura interna de las lecturas de los sentidos evaluando aspectos como su simetría, armonía, regularidad, etc. Lo percibido sería finalmente aquello que *mejor* se adecuase a estos aspectos.

Teoría de *sense-data*. Esta aproximación al fenómeno es un paso intermedio entre la teoría de percepción directa y las demás teorías de percepción mediada. El concepto de *affordance*, característico de la percepción directa, es aquí *sense-datum*, que podría traducirse aproximadamente como *dato sensorial*.

Un *sense-datum* es lo que el sistema percibe directamente del entorno. A diferencia de la percepción directa, esta teoría admite que un *sense-datum* depende del sistema y de su estado.

Teoría de visión de Marr. [Mar82] se ha convertido en un texto clásico sobre visión. Se puede resumir en tres puntos [Mar82, p.329-332]:

- Representación y procesamiento tienen una naturaleza dual y son mutuamente dependientes en los sistemas.
- La visión no puede ser explicada con un único nivel de abstracción. Requiere estudiar tres aspectos que forman un todo: el aspecto computacional, el de algoritmo y el de implementación.
- En visión existen tres tipos de representaciones dependiendo de la perspectiva y el nivel de abstracción que se considere: (1) *primal sketch*, relativo al análisis de la estructura de la lectura sensorial, (2) el modelo $2\frac{1}{2}$–D (dos dimensiones y media) en el cual los objetos del entorno se representan desde el punto de vista del observador y (3) el modelo 3-D (tres dimensiones), por el cual se representan como conceptos en sí, independientes del observador.

Esta aproximación al fenómeno incide en múltiples aspectos aún no resueltos de la percepción: percepción multirresolución, dependencia del contexto, representación del conocimiento, etc.

Percepción Indirecta. Esta visión —que da nombre a toda esta línea de teorías— deriva de la visión de Helmholtz [Roc85], [Roc97], [Roc83], [Ull80]. La generaliza al concebir la percepción como un proceso puramente de resolución de problemas, cuyo fin no es necesariamente deducir la explicación más probable de una lectura sensorial.

Todas las corrientes dentro de esta aproximación consideran la influencia de aspectos como la experiencia pasada del observador, su estado y su conocimiento.

Conviene considerar la visión aportada en [Sha05], que ofrece una formalización de esta aproximación y que resume múltiples aspectos críticos relacionados.

Embodied Cognition. También se llama *situated cognition*. Se puede considerar la tendencia actual. Para una visión general, se puede consultar [And03a], [Chr03], [And03b]. Parte de la base de que la cognición —entendida como un fenómeno más general que la percepción— es un fenómeno situado (*situated*) entendiendo por ello que no puede ser explicado aisladamente de un contexto más amplio.

Está basado fundamentalmente en la evidencia que muestra la estrecha relación entre fenómenos y partes del sistema anteriormente considerados independientes, como la percepción y la acción, o los centros motores del cerebro con los perceptivos.

Este campo se enfrenta a diversas cuestiones críticas. Podemos resaltar:

- Integración de las operaciones del sistema: comportamiento coherente, dependencias entre procesos —perceptivos y otros—, órganos y elementos del sistema. Principios de funcionamiento subyacentes a la cohesión del sistema (homeostasis, emociones).

- *Corporización*. Dependencia entre cognición y aspectos estructurales y de substrato del sistema: capacidades, restricciones y características derivadas del substrato.
- Funcionalidad: Relación entre la cognición y los objetivos del sistema. Relación con aspectos fundamentales del sistema como la homeostasis.

2.2 Tesis

Se expondrá la tesis fundamental de este trabajo en cinco puntos. Las siguientes secciones aportarán mayor detalle y desarrollarán el contexto en que deben entenderse.

I. Sobre el proceso perceptivo. Todo proceso perceptivo implica tres aspectos: *estímulo cercano*, *singularidades* y *objetos*.[1] El proceso perceptivo consiste en relacionar los tres aspectos.

El término 'estímulo cercano' se refiere a los valores de la magnitud medida por un sistema sensorial, en la barrera de éste con su entorno. Es decir, antes de que pudieran ser alterados por el sistema en modo alguno.

Globalmente, la percepción siempre sigue una secuencia de dos fases a la que se llamará *secuencia fundamental*, representada esquemáticamente en el siguiente diagrama:

SP: procesamiento sensorial / de información cercana

DP: procesamiento dirigido / de información cognitiva

La terminología de este trabajo está escrita en cursiva en la figura. Otras terminologías usuales se indican con letra sin serifa.

SP y *DP* representan las dos fases de la secuencia fundamental.

El proceso perceptivo está orientado a reconocer objetos específicos en el entorno con detalle y atribuciones propias de cada uno (relevancia, asociaciones, connotaciones), ignorando otros. Los objetos a cuyo reconocimiento está orientado el proceso se llamarán *referentes* del proceso. Esta orientación puede ser implícita

[1] El término 'objeto' debe entenderse como *concepto*, *idea* o *entidad conceptual*.

o explícita, dependiendo de si el proceso perceptivo manipula representaciones simbólicas de estos objetos o no.

Las *singularidades* son patrones en los valores del estímulo cercano.[2] Estos patrones son atribuidos —por el subproceso *DP*— a una configuración concreta de los objetos del entorno. Es esta configuración la que se representa como un conjunto de *objetos percibidos*.

Es decir, DP consiste en establecer relaciones de equivalencia entre los referentes y partes del entorno que interpretará como objetos. Es decir, una parte del entorno que presenta una forma y un cuerpo concretos, puede ser considerada —según su forma y su cuerpo— como una *instancia* de un *referente*. Una forma *real* del concepto abstracto que representa el *referente*.

En resumen: DP interpreta el análisis de singularidades llevado a cabo por SP. De él deduce: (1) si existen instancias de sus referentes en el entorno —objetos— o no. (2) En caso de existir, la forma concreta que presentan, es decir: sus *atributos* concretos.

II. Sobre el contexto de la percepción. El proceso de la percepción depende del sistema y su entorno. La percepción recibe influencia del resto de los procesos del sistema. A su vez, ejerce influencia sobre el resto del sistema. Esto puede ocurrir de dos formas: primero, a través del efecto explícito que pudieran causar los objetos percibidos. Segundo, modificando el sistema —implícitamente— en su transcurso.

Los conceptos introducidos en el capítulo dedicado a sistemas desarrollan el contexto en el que tiene lugar la percepción: múltiples procesos perceptivos corporizados en recursos que pueden ser compartidos o dependientes de otros, relación entre procesos perceptivos y comportamiento, organización y objetivos entre otros. Estos aspectos determinan en gran medida el propósito, el desarrollo, las capacidades y la importancia relativa de un proceso perceptivo.

III. Relevancia cognitiva. La percepción es un proceso dirigido al reconocimiento de instancias de objetos. Los objetos en sí forman parte del conocimiento del sistema. Se llaman *referentes* porque el proceso está referido a ellos.

Los referentes determinan el *punto de vista* de un proceso perceptivo: aquello que es interesante y aquello que no (y en qué medida y forma). Es decir, determinan la *finalidad* del proceso.

Los *objetos percibidos* que resultan de un proceso de percepción son necesarios para los procesos de resolución de problemas, planificación y monitorización del estado del sistema entre otros. Aportan el nexo entre el mundo real y la operación del sistema.

Las operaciones con referentes y objetos percibidos[3] son necesarias para simular escenarios de operación hipotéticos, para refinar algoritmos y procesos, y para ampliar el conocimiento. Son necesarias para crear nuevos referentes.

[2]Este concepto generaliza múltiples nociones que pueden encontrarse en la literatura >*p.189*.

[3]Por ejemplo: generalización, asociación, particularización.

IV. Dominio de percepción. Un proceso perceptivo percibe sobre una parte del universo que llamaremos *entorno perceptivo*. Incluye el mundo exterior al observador, que llamaremos *entorno del sistema*, y también —en el caso general— partes del propio sistema. Es decir, que un mismo proceso perceptivo puede percibir dentro y fuera del sistema.>*fig.10.2, p.171.*

Percibir dentro o fuera del sistema —o ambos— es irrelevante en cuanto a la estructura y naturaleza de un proceso perceptivo.

Puede haber diferencias, sin embargo, en cuanto a la corporización que implica; al nivel de procesamiento asociado o cualquier otro aspecto específico de un sistema concreto. Por ejemplo, procesos que perciben dentro de sistemas biológicos generalmente disponen de *estímulos cercanos* más ricos en cuanto a la naturaleza y el número de estímulos, debido a la densidad de terminaciones nerviosas (riqueza de las lecturas).

V. Sobre la interacción de SP y DP. Ambos tipos de procesamiento pueden interaccionar y ejercer mutua influencia en el transcurso de un proceso perceptivo.

2.3 Percepción

En esta sección vamos a construir una visión general de la percepción. Introduciremos conceptos básicos sobre las partes que intervienen en el proceso así como sobre la dinámica del proceso en sí. Esto completará la visión de la percepción desde el punto de vista del sistema que se inició en el capítulo anterior.

En primer lugar, analicemos las partes que intervienen en el proceso. A primera vista, podemos distinguir el *sistema*, al que también llamaremos *observador*, y su entorno, al que llamaremos *entorno del sistema*. A grandes rasgos, el observador percibirá sobre su entorno. Esta es la noción más intuitiva sobre la percepción. >*fig.10.1, p.170*

Conviene recordar, sin embargo, una consideración derivada de la GST. La frontera entre el sistema y el entorno es difícil de distinguir, y en la mayoría de los casos reales, imposible. Por lo tanto, la división entre observador y entorno no es clara. Sin embargo, la noción intuitiva debe aceptarse como la noción básica.

No todo el sistema percibe. Hay una parte de él dedicada específicamente al proceso, mientras que el resto puede estar dedicado a otras tareas o en reposo. La parte que percibe consistirá en un conjunto de recursos especializados. Nos referiremos a esta parte del sistema como *perceptor*.

Cabe plantearse la noción básica del proceso de nuevo. ¿El perceptor percibe sobre el *entorno del sistema* o sobre lo que le rodea a él como elemento? En general, podemos asumir que percibe en torno a él mismo. Es decir, sobre *su* entorno, al que llamaremos *entorno perceptivo*. Observamos que el *entorno perceptivo* incluye en general, además del *entorno del sistema*, partes del sistema en sí. >*fig.10.2, p.171*

El perceptor es una parte del sistema y por lo tanto presentará acoplamientos, dependencias y restricciones con los demás elementos. La operación del perceptor es, en suma, una sucesión de cambios en los valores de sus cantidades. Éstas, estarán relacionadas con las del resto del sistema a través de las correspondientes *relaciones de comportamiento*. Por tanto, la mutua influencia perceptor-sistema ocurrirá a lo largo de todo el proceso perceptivo, y por supuesto al concluir con el *objeto percibido*.

Esta mutua influencia no debe olvidarse al analizar la percepción, ya que representa realmente el contexto en que ésta tiene lugar. Los resultados del proceso pueden variar en función de las influencias que hayan ocurrido durante el proceso perceptivo, y el propio sistema puede verse también afectado durante el transcurso del proceso. La forma en que esta mutua influencia ocurra es parte de la directividad del sistema: objetivos, estructura funcional, etc.

2.3.1 Visión general de la percepción

De acuerdo a los conceptos introducidos previamente, la *percepción* puede concebirse como un proceso que produce cambios en el sistema, relacionados no-aleatoriamente con el estado del *entorno perceptivo*.

Como se apuntaba anteriormente, de hecho estos cambios tienen un significado relativo a la *finalidad* del sistema. Los cambios derivados del proceso son una representación del estado del *entorno perceptivo*. Esta representación no es en general objetiva, sino que es *relativa al sistema*: objetivos, recursos, estado, etc.

A bajo nivel, el *objeto percibido* es una representación construida con cantidades del sistema. Los valores de estas cantidades serán, por tanto, *conceptuales*, al referirse a cantidades reales del entorno. El esquema básico de la percepción es el siguiente:

$$\text{estímulo cercano} \xrightarrow{\text{percepción}} \text{objeto percibido}$$

Volvamos al concepto de *perceptor* introducido previamente, para analizar este esquema en más detalle. Podemos distinguir tres fases de la percepción:

1. El entorno producirá ciertos cambios en las cantidades del perceptor.
2. Siguiendo las dependencias mutuas entre estas cantidades y las del resto del perceptor, se inducirán cambios en las segundas.
3. Finalmente, esta sucesión de cambios entre las cantidades del perceptor concluirá en las cantidades del acoplamiento perceptor-sistema.

Los cambios de la fase 3 son la representación que veníamos denominando *objeto percibido*. Sin embargo, no olvidemos que el perceptor también puede inducir cambios en el sistema durante la fase 2. A estos cambios los llamaremos *cambios implícitos*.[4]

Todos los cambios generados por el proceso perceptivo, explícitos o implícitos, obedecen al contexto sistémico en que se desarrollan. Como vimos en el capítulo anterior,

[4]Tampoco debemos olvidar que el sistema puede influir al proceso perceptivo en esta fase.

la percepción tiene lugar como la parte *aferente* de una función que se corresponde con un objetivo del sistema.

Podemos concluir de este análisis que la percepción tiene dos valores para la operación del sistema. Explícitamente, proporciona representaciones. Implícitamente, genera cambios en la organización del sistema. Intuitivamente:

- Un valor de información, el conocimiento explícito derivado.

- Un valor estructural o de substrato, afectando a la operación de los demás elementos del sistema, como en el caso de las emociones humanas.

Llamaremos genéricamente *percepto* al conjunto de cambios derivados del proceso perceptivo. Por tanto, un percepto tiene valor de información y valor estructural.

2.3.2 Perceptor

Podemos concluir de todo lo anterior que los acoplamientos del perceptor con el resto del sistema son esenciales para la percepción. Por una parte son la entrada al sistema de información relativa al entorno perceptivo. Por otra parte son la vía por la cual la operación del perceptor se integra con la del resto del sistema, a través de los cambios implícitos. Analicemos pues las partes del perceptor basándonos en las tres fases descritas anteriormente. >*fig.10.5, p.177*

Durante la primera, el entorno induce cambios en un conjunto de cantidades del perceptor que llamaremos *sistema sensorial*, SS. Podemos deducir que las cantidades del sistema sensorial serán en su mayor parte *cantidades independientes*.[5] Estas cantidades determinan el *estímulo cercano* en el cual se basará todo el proceso perceptivo.

Durante la segunda fase, las dependencias del sistema sensorial con el resto del perceptor inducen cambios en éste. Podemos distinguir dos casos. Primero, que las cantidades en las que se induce los cambios pertenezcan al acoplamiento del perceptor con su entorno perceptivo. Segundo, que no formen parte de un acoplamiento. En este segundo caso, las llamaremos *cantidades interdependientes*, ID.

Las cantidades del acoplamiento del perceptor con el entorno pueden categorizarse más —ya habíamos distinguido las del sistema sensorial—. El *objeto percibido* se trasladará al sistema a través de una parte del acoplamiento que llamaremos *sistema de representación*, RS. Habrá una tercera parte del acoplamiento a la que llamaremos *acoplamiento implícito*, IC. Esta parte, a su vez, podrá dividirse en dos: una que esté acoplada al *sistema* y otra al *entorno del sistema*. Nos referiremos a la primera como *acoplamiento de sustrato*, SC, y a la segunda, *acoplamiento marginal*, MC. Los cambios derivados del acoplamiento implícito (MC + SC) tendrán una influencia implícita sobre el proceso perceptivo, como se ha apuntado en secciones anteriores.

[5]Es decir: impuestas por el entorno.

2.3.3 Dinámica perceptiva

Las secciones anteriores han analizado la percepción desde el punto de vista de las partes que intervienen en ella. En ésta, vamos a analizar las fases generales de un proceso perceptivo.

Desde un punto de vista cognitivo, la actividad de un perceptor consiste en ejecutar operaciones. Éstas pueden ser: analogía, reconocimiento, asociación, generalización, etc. A menor nivel de abstracción, todas estas operaciones equivalen a sucesiones de cambios en los valores de las cantidades del perceptor.

No todos los procesos perceptivos son iguales respecto a nivel de abstracción, recursos empleados o complejidad. Eso hace que exista una gran variedad de operaciones perceptivas. Pueden ser diferentes a nivel cognitivo, como por ejemplo *reconocimiento de objetos* frente a *generalización*, o pueden implicar subprocesos de otra naturaleza: *eferentes* o *deliberativos*.

Como se ha mencionado previamente, la percepción establece una equivalencia entre sus *referentes* y los objetos del entorno, que representa como *objetos percibidos*, a través del *estímulo cercano* y de las *singularidades*. Los subprocesos perceptivos involucrados se pueden clasificar en dos grandes categorías:

- Procesos de adquisición y ajuste operativo del sistema sensorial.
- Procesos que transforman los valores producidos por el sistema sensorial siguiendo una finalidad concreta.

Llamaremos genéricamente *procesamiento sensorial* o bien *de información cercana* a los primeros. Llamaremos *procesamiento dirigido* o *de información cognitiva* a los segundos. En función de estos procesos, podemos concebir la percepción como una secuencia de dos fases —*secuencia fundamental*—:

$$\text{percepción sensorial} \longrightarrow \text{percepción dirigida}$$

Esta secuencia se cumple en todo proceso perceptivo, aunque en procesos reales y complejos se cumplirá solo a nivel conceptual. En general, un proceso perceptivo con cierto grado de complejidad implicará múltiples subprocesos que se darán en diversas formas: serie, paralelo, iteraciones, repeticiones, etc. También puede implicar subprocesos eferentes o deliberativos coordinados con los demás.

Por tanto, podemos concluir que la percepción, en general, puede involucrar múltiples subprocesos que pueden intercambiar resultados intermedios, compartir recursos, coordinarse, etc. La forma en que se combinen dependerá de multitud de factores, entre los que se cuenta la influencia implícita tanto del entorno como del resto del sistema a través de IC. Por tanto, aunque siempre haya unos procesos de tipo sensorial y otros de tipo dirigido y en conjunto se siga una secuencia semejante a la fundamental, a nivel individual las combinaciones pueden ser muy distintas. *>fig.10.6, p.179*

2.3.4 Memoria perceptiva

En términos cognitivos, el valor principal de la percepción se encuentra en la representación, es decir, en el *objeto percibido* y su valor de información.

El objeto percibido, a bajo nivel, consiste en combinaciones y valores de un subconjunto de las cantidades del sistema. Un sistema puede disponer de ciertos recursos especializados para estas representaciones: capaces de modificar su valor rápidamente, adaptados a las necesidades del proceso perceptivo, que no impongan excesivas restricciones que pudieran alterar el valor de información de las representaciones, con suficiente capacidad, etc. Los llamaremos genéricamente *recursos de representación* del sistema.

De ellos, en general solo una parte estará asociada a la operación de un perceptor en concreto. Llamaremos a esta parte *memoria perceptiva*. Por tanto, la memoria perceptiva asociada a un proceso de percepción contiene el conocimiento explícito generado por él.

Las representaciones contenidas en la memoria pueden tener diversos niveles de persistencia. Aquellos que potencialmente pudieran ser relevantes a los objetivos del sistema tendrán en general mayor persistencia que otros asociados a objetivos de menor nivel o a operaciones intermedias, que podrán ser eliminados por otros procesos perceptivos o simplemente desaparecer.

2.3.5 Percepción distribuida

Como mencionamos en el capítulo dedicado a sistemas, la operación de los sistemas autónomos puede entenderse en términos de una estructura funcional que se corresponde a una jerarquía de objetivos. La estructura funcional, a su vez, puede expresarse mediante *nodos*, y cada nodo representaría a una función de la estructura. Como hemos mencionado previamente, la percepción constituye una parte de cada nodo.

Este modelo, en conjunto, representa por tanto un sistema paralelo y distribuido, en el que la operación del conjunto se modeliza a través de un conjunto de procesos que pueden tener lugar en diferentes partes del sistema. En este contexto la percepción del sistema no es, por tanto, un proceso único, sino un conjunto distribuido de procesos, cada uno asociado a un nodo: recursos, objetivos, procesos deliberativos y aferentes, nivel de resolución espaciotemporal, etc.

2.4 Sistemas perceptivos

En la sección anterior se ha construido una visión general de la percepción, y se ha identificado sus partes principales. Procedamos ahora a analizar aspectos relativos a los tipos de percepción que se derivan del marco anterior y a las fases de procesamiento de información que implican.

Podemos concluir de las secciones anteriores que el perceptor está formado por cinco partes que hemos llamado *sistema sensorial*, SS, *sistema de representación*, RS, *cantidades* *interdependientes*, ID, *acoplamiento marginal*, MC, y por último *acoplamiento de sustrato*, SC.

El *objeto percibido* se traspasa al sistema a través de RS. A este aspecto de la percepción le llamaremos *percepción cognitiva*. Los *cambios implícitos* se inducirán en el sistema a través de SC. Esto lo denominaremos *percepción corporizada*. Estos dos valores de la percepción ya se habían comentado en las secciones anteriores.

Sin embargo, pueden darse más tipos de percepción. Recordemos que un *sistema perceptivo*, entendiendo por tal al conjunto de elementos directamente involucrados en la percepción, consta fundamentalmente de perceptor y memoria, y éstos se relacionan indirectamente con el entorno y el resto del sistema. >*fig.11.2, p.11.2*

La interacción de RS con la *memoria perceptiva*, PM, se denominará *representación*. El perceptor, como se mencionó previamente, puede percibir sobre el entorno del sistema y sobre partes del sistema en sí. Denominaremos este segundo caso *propiocepción*. En el caso de que perciba sobre la memoria perceptiva, entenderemos que es *metapercepción*.

Conviene resaltar que un proceso perceptivo puede estar basado en más de un referente; es decir, que puede reconocer más de un objeto en el entorno. En general, estará basado en un conjunto de referentes. Incluso puede ocurrir que para reconocer algunos sea necesario reconocer otros previamente. La operación de un perceptor, por tanto, puede estar compuesta de procesos perceptivos más simples.

Entenderemos que un *perceptor* es un conjunto coordinado de *sentidos*. Un *sentido* es una parte del perceptor que está especializada en el reconocimiento de un referente concreto. En las secciones siguientes analizaremos cómo es la percepción de un *sentido*.

2.4.1 Sentidos

Un sentido realiza un análisis selectivo del *entorno perceptivo* desde un *punto de vista* concreto. En particular, desde el punto de vista que le corresponde dentro de la estructura funcional del sistema. Un punto de vista, formalmente, consiste en [Kli69, p.39], [Kli80, p.73]:

- Nivel de resolución.
- Un conjunto de cantidades consideradas.
- Relaciones de comportamiento entre las cantidades.
- Las propiedades que determinan estas relaciones.

El *nivel de resolución* define la operación del sentido en el espacio y en el tiempo. Podemos considerar que este aspecto deriva fundamentalmente de la implementación, y que por tanto tiene que ver con la *corporización* concreta del sentido. Viene determinado, por una parte, por la capacidad de los recursos en los que está corporizado el sentido, y por otra parte, por el resto del sistema (compartición de rescursos, etc.). A pesar de que las limitaciones de resolución provengan fundamentalmente del sustrato, puede tener influencia sobre el valor cognitivo de la percepción.

El *referente* del sentido, determina los aspectos del punto de vista relativos a la información, es decir, el resto: el conjunto de cantidades que el sentido evaluará, las relaciones entre ellas y las propiedades que justificarán sus valores. En las secciones siguientes analizaremos en más detalle el procesado de información.

Procesamiento de información cognitiva, DP

Como todo proceso perceptivo, el proceso que lleva a cabo un sentido se ajusta a la *secuencia fundamental* introducida previamente.

En un *sentido*, el procesamiento de información cognitiva procesa *singularidades*,[6] y de ellas infiere un cierto estado del entorno perceptivo. En concreto, infiere el estado de los objetos del entorno que se corresponden con el *referente*. El resultado es el *objeto percibido*.

El proceso de inferencia consiste en establecer una equivalencia entre el conjunto de singularidades observado, $\Psi = \{\psi_j, \quad j = 1..n_\Psi\}$, y el referente del sentido, ρ. Podemos representar esta equivalencia como una relación ϵ, de forma que:

$$\rho^R = \epsilon(\Psi)$$

A la relación ϵ se le llamará *equivalencia cognitiva*. El superíndice 'R' indica que se trata de una *representación* de una instancia del referente. A esto es a lo que llamamos *objeto percibido*.

Como se mencionó arriba, un perceptor, en el caso general, puede integrar múltiples sentidos, y percibir, por tanto, un conjunto de referentes, $\mathbf{V} = \{\rho^i, \quad i = i..n_\rho\}$. La operación del perceptor se podrá expresar, por tanto, con un número igual de relaciones de equivalencia:

$$\rho^{iR} = \epsilon^i(\Psi^i), \qquad i = 1..n_\rho$$

Esta formulación es sólo válida, en general, conceptualmente, ya que pueden existir relaciones de equivalencia que dependan de otros objetos percibidos, y por tanto, no de singularidades directamente como en la formulación de arriba. Pueden ocurrir también casos mixtos en los que la ϵ dependa tanto de objetos percibidos como de singularidades.[7]

Procesamiento de información cercana, SP

En síntesis, el procesamiento de información cercana se refiere a todos los procesos que intervienen en calcular *singularidades* a partir del *estímulo cercano*. Podemos distinguir

[6] También puede basarse en *objetos percibidos* previamente por otros sentidos coordinados con él, o guardados en la *memoria perceptiva* asociada al proceso.

[7] Siempre se puede representar la percepción de un referente como en la formulación de arriba, en función de sus singularidades. Sin embargo, puede ocurrir que el proceso perceptivo no esté implementado así —partiendo directamente de sus singularidades— sino que se base *objetos percibidos* intermedios o en mezclas de objetos percibidos y singularidades. Un ejemplo de este caso se puede ver en el capítulo dedicado al sistema DAM 12.

dos tipos de proceso:

Equivalencia de singularidad: Se entiende por *equivalencia de singularidad* al cálculo de un conjunto de singularidades a partir de los valores de las cantidades del sistema sensorial.

Podemos indicar el conjunto de singularidades considerado como $\Psi = \{\psi_k, \quad k = 1..n_\Psi\}$ y el conjunto de cantidades de SS empleadas para calcular cada una como Q_k. El procesamiento de información cercana, entonces, calculará n_Ψ relaciones de equivalencia de singularidad, σ_k:

$$Q_k \xrightarrow{\sigma_k} \psi_k, \qquad \psi_k = \sigma_k(Q_k)$$

Ecualización: Entenderemos por *ecualización* la modificación del valor de una cantidad de entrada a un proceso de *equivalencia de singularidad*, respecto a la correspondiente del *sistema sensorial*. Es decir, indicando el valor de la cantidad k-ésima del sistema sensorial como q_k^{ss}, la ecualización equivale a:

$$q_k^{ss} \xrightarrow{\quad eq_k \quad} q_k^{ss*}, \qquad q_k^{ss*} = eq_k(q_k^{ss})$$

En donde eq_k indica el proceso de ecualización de la cantidad k-ésima y el asterisco indica el cambio de valor. El valor ecualizado, q_k^{ss*} sería la entrada al proceso de *equivalencia de singularidad*, en lugar del original, q_k^{ss}.

La ecualización puede tener dos motivos. Primero, mejorar los valores de q_k^{ss}, corrigiendo posibles desviaciones debidas al sustrato ('errores de lectura' en sistemas artificiales). Segundo, adaptar el *proceso de información cercana* al punto de vista del *sentido*. Por ejemplo, amplificando los valores de cantidades importantes para el proceso, o de cantidades que puedan indicar riesgo de fallo en el sistema,[8] etc.

[8] Fallo de programa, fallo estructural.

Chapter 3

Conclusiones y trabajo futuro

Este trabajo ofrece un marco conceptual básico para sistemas autónomos generales. Este es un punto de vista muy amplio que cubre múltiples aspectos de los sistemas. Se adoptó con el fin de alcanzar una comprensión unificada y general.

Este marco está formado por conceptos muy abstractos heredados o generalizados a partir de estudios previos. Muchos de estos conceptos no se habían integrado antes en un marco sistémico común, bien porque surgieron de disciplinas muy dispares, o bien porque las diferencias entre sus niveles de abstracción o campos de aplicación los hacían parecer totalmente independientes.

3.1 Recapitulación de Objetivos

I. Generalidad. Fue un primer objetivo de este trabajo. Finalmente se ha convertido en una necesidad. Debido a la heterogeneidad de los sistemas, para poder analizarlos desde un punto de vista común es necesario situarse en un nivel muy general. Entendemos que este objetivo se ha alcanzado, y que en parte esto ha sido gracias a la adopción de la Teoría General de Sistemas como base.

II. Obtención de conceptos, principios y relaciones, de aplicación a la ingeniería de sistemas. Esto fue un segundo objetivo. Concluimos que la ontología que resulta de este trabajo incrementa el conocimiento actual, ya que identifica las nociones y los principios fundamentales que explican los sistemas y su operación. La ingeniería se beneficia de esto con una nueva perspectiva de los sistemas, que mejorará su comprensión. Sin embargo, consideramos que este objetivo se ha alcanzado solo en parte, puesto que la aplicabilidad de este trabajo es aún limitada:

- La aplicación de los principios de diseño, explícitamente mencionados en el trabajo o derivados de él, es posible a un nivel cualitativo. Esto puede ser útil para el ingeniero aportándole una visión coherente de la relación entre autonomía, propiedades del sistema y percepción entre otros aspectos. El ingeniero puede explotar esta visión para la fase de diseño general o preliminar, pero aún no puede aplicarse a la ingeniería de detalle (análisis, síntesis).

- La complejidad de los sistemas resulta un aspecto importante, dado que algunos conceptos mencionados aquí pueden cambiar su forma significativamente en función de ella, como se ha apuntado a lo largo del texto, por ejemplo al hablar de la posibilidad de elementos *degenerados*. Es necesario, por tanto, un desarrollo teórico para relacionar los conceptos expuestos aquí con la complejidad de los sistemas. También una metodología sistemática de aplicación basada en la complejidad. Un punto de partida para estos desarrollos puede encontrarse en los principios de simplificación mencionados en [Kli01, p.159-170].
- La aplicabilidad de este trabajo no está limitada solo por la complejidad. Aunque se proporciona una colección de casos de aplicación, es necesario desarrollar una metodología sistemática que en este momento no existe. En la situación actual existe indeterminación respecto a la forma de aplicar este trabajo a sistemas reales. A fecha de hoy, se piensa que esta metodología debería ser iterativa y progresiva. También se piensa que debería basarse en una colección inicial de aspectos sistémicos que fuese progresivamente ampliada durante el proceso de análisis o síntesis. Esta metodología debería concretar criterios para determinar los aspectos sistémicos fundamentales de partida y para diseñar los procesos de iteración, en función del entorno, los recursos y los objetivos del sistema. A fecha de hoy, estos puntos están a criterio del ingeniero.

Debe resaltarse que un aspecto crucial para la aplicabilidad de este trabajo a los sistemas artificiales —análisis y síntesis— es la introducción de los *objetivos* del sistema dentro de la relación de causalidad heredada de [Kli69], dándoles una importancia teórica explícita. De hecho, esto conduce a una de las relaciones principales en las que se basa el trabajo:

$$\text{objetivos} \longrightarrow \text{propiedades (organización)} \longrightarrow \text{comportamiento}$$

En donde los *objetivos* representan los propósitos que *dirigen* la evolución del sistema; implícitos o explícitos, de corto o de largo plazo, propios del sistema o impuestos por el diseñador.

El trabajo explica los sistemas y la percepción incluyendo los objetivos como algo *intrínseco*, constitutivo de los sitemas, como lo son su masa, longitud o propiedad en general. Consideramos que, aunque la importancia de los objetivos en los sistemas ha sido entendida previamente,[1] este trabajo constituye un primer intento de analizar los objetivos como una parte integral de los sistemas generales, más allá de implementaciones y arquitecturas concretas.

[1] Se puede consultar la sección referente a la *finalidad* en sistemas generales. Desde luego, los objetivos son un factor de diseño crítico en los sistemas artificiales: los objetivos del diseñador se incorporan a los sistemas. Las arquitecturas guiadas por objetivos —ver el estado del arte de los sistemas autónomos— son ejemplos en los que el sistema puede manejar objetivos explícitamente.

Este punto precisamente abre otra línea de investigación: el desarrollo de metodologías que sistematicen diseño orientado a objetivos integrando objetivos, organización y comportamiento. Un ejemplo de una metodología orientada a objetivos en un dominio concreto es el conocido método de diseño del *lugar de las raíces* para sistemas de control [Oga90].

III. Glosario. El objetivo de construir un glosario de términos de percepción surgió durante el desarrollo de este trabajo. El progreso llevado a cabo puede consultarse en la parte VI. Consideramos que este objetivo solo se ha alcanzado parcialmente. El glosario incluido aquí requiere revisarse en profundidad y ampliarse con más términos.

Sin embargo, la experiencia ha llevado a ciertas conclusiones:

- Incluir adecuadamente los términos del presente trabajo requiere definir también otros relacionados. Esto puede complicar la tarea considerablemente.
- Definir un término puede requerir en ocasiones explicaciones muy extensas. Esto se puede simplificar mediante la adición de segundas acepciones y significados en contextos diferentes, que ayuden al lector a construirse una noción global.
- Para incluir términos de muy alto nivel de abstracción como los del trabajo actual es más práctico separarlos de los términos específicos de un dominio en glosarios independientes.

3.2 Trabajos futuros

Entendemos que las relaciones y los conceptos de este trabajo se consolidarán tras su inmersión sostenida en la dinámica científica. La consolidación de esta propuesta teórica consiste en dos puntos:

- Refinar el alcance, la precisión y las relaciones de los conceptos.
- Construir una casuística de aplicación completa.

El segundo punto equivale a la experimentación y a la observación de posibles imprecisiones, omisiones y contradicciones, proceso necesario para refinar los conceptos. También constituiría un cuerpo de conocimiento en el que basar metodologías de aplicación. Consideramos que este aspecto es esencial.

Teniendo en cuenta todas estas consideraciones, vemos las siguientes líneas de progreso para el futuro inmediato:

Formalización: Es necesaria para hacer posibles procesos sistemáticos de ingeniería, modelización y razonamiento con los conceptos de este trabajo. En segundo lugar, constituye una herramienta para refinarlos, al proporcionar un método no ambiguo de representación conceptual.

Inicialmente, la formalización se planteó como parte del trabajo. Para ello se exploró diversas áreas: geometría, matemáticas, representación del conocimiento y modelización de software entre otros. De ello se concluyó que la formalización de este trabajo es en sí misma materia de una investigación.

Como se ha mencionado arriba, uno de los objetivos de este trabajo ha sido relacionar aspectos de comportamiento de los sistemas (externos) con otros relativos a objetivos y organización (internos). Se tiene un interés especial en encontrar expresiones formales para estas relaciones. Esto resulta especialmente difícil con algunos de los conceptos más importantes de este trabajo. Estructura mínima, homogeneidad, encapsulación, isotropía del conocimiento y escalabilidad son ejemplos de ello. Estos conceptos nos ayudan a entender los sistemas, pero son difíciles de expresar formal y cuantitativamente, y sin ambigüedad. Vemos dos posibles caminos para lograrlo:

- *Teoría de categorías* [LS97] [Pie91]. Podría ser una herramienta de modelización y formalización dado su tratamiento sistemático de *conjuntos* y *morfismos*. Debemos tener en cuenta que la teoría general de sistemas, en la que se basa este trabajo, se concentra más en las relaciones entre las cosas que en las cosas en sí,[2] en línea con la necesidad de formalizar morfismos.
- Modelos ejecutables. Algunas herramientas permiten implementar sistemas software a partir de modelos conceptuales, expresados por lo general en UML o lenguajes similares. Tenemos la intención de construir modelos que sigan la ontología propuesta aquí, aplicados a sistemas concretos, con el fin de construir una casuística y refinar los conceptos.

Investigación. Existen múltiples aspectos relativos a los sistemas que han sido identificados aquí de los que prácticamente no existe un conocimiento consolidado, y que aún no pueden ser implementados, o solo hasta cierto punto. La investigación es necesaria para teorizar sobre ellos y poder ser incluidos en esta ontología.

Un ejemplo significativo de la necesidad de conocimiento es el *principio de escalabilidad*. Ignoramos la existencia de metodologías o principios generales de diseño de sistemas para conseguir propiedades invariantes con la escala del sistema. Hemos identificado, sin embargo, dos temas de interés en relación con ésto: la teoría de fractales [Man00], y la geometría y el estudio del crecimiento [Coo14], [Ghy83], [Tho61]. En ambos casos, es necesario establecer los isomorfismos adecuados entre sus conceptos y los propuestos aquí.

Aplicación: Metodologías, Análisis, Sintesis. Consideramos que es esencial aplicar este trabajo a sistemas concretos. Esto implica diseñar metodologías y experimentos. Como se mencionó anteriormente, el objetivo de esto debe ser construir una casuística que incluya sistemas heterogéneos en complejidad y naturaleza.

[2] Consultar el primer capítulo de [Kli01] para una explicación del alcance y el propósito de la teoría general de sistemas.

Como hemos mencionado, estamos actualmente investigando la posibilidad de aplicar modelos ejecutables a sistemas reales. Se está considerando por el momento un robot móvil, una planta de proceso continuo y un sistema software.

3.3 Una teoría unificada de percepción

Uno de los objetivos principales de este trabajo fue construir una teoría **unificada** de percepción, que recogiese las aportaciones y descubrimientos de las aproximaciones anteriores, y que fuese de ámbito general, con el objeto de poder aplicarse a cualquier tipo de sistema. Elaboremos una perspectiva frente a otras teorías de percepción:

Percepción abductiva. Tal vez el planteamiento del problema de la percepción más cercano a este trabajo es el descrito en [Sha05]:

- El punto I de esta tesis, relativo al proceso perceptivo y la *secuencia fundamental* se comparte, en esencia, entre ambas teorías. Ambas asumen un cierto grado de *procesamiento de información cercana* y una fase de *procesamiento de información cognitiva*.
- El papel de las *singularidades* también se identifica en ambas teorías no necesariamente como una descripción del mundo exterior, sino del estado de los sensores (*sistema sensorial* en este trabajo). Esto implica la existencia de una cierta *equivalencia cognitiva* que debe ser establecida por los procesos de inferencia (*procesamiento de información cognitiva* en este trabajo).
- La noción de *umwelt* también se aborda en términos similares en ambos casos.
- En los dos casos se asume que la percepción implica flujos de información tanto *bottom-up* como *top-down*.

Sin embargo, se dan diferencias significativas:

- Este trabajo aporta mayor detalle relativo a las operaciones de *procesamiento de información cercana*: *ecualización* y *equivalencia de singularidades*, no analizados en [Sha05].
- Este trabajo desarrolla un marco conceptual sobre sistemas autónomos generales, que proporciona una visión detallada del contexto en que tienen lugar los procesos perceptivos: nodos, estructura funcional, objetivos, finalidad, etc.

  A pesar de que [Sha05] trata la fusión sensorial, que implica múltiples procesos perceptivos, esto representa un caso particular, que no aporta información sobre aspectos sistémicos: la relación de la percepción con procesos eferentes y deliberativos, descomposición funcional, directividad, etc.
- Los flujos de información *top-down* se tratan solamente en el caso particular de 'predicciones' —*expectation*—. Se entienden como un mecanismo heurístico, involucrado en la fase de *procesamiento de información cognitiva*.

De acuerdo con este trabajo, existen múltiples mecanismos de flujo *top-down*. Ejemplos pueden ser: percepción implícita, *re-sensing* y el caso (c) ilustrado en la sección de dinámica perceptiva (consultar versión inglesa). Aparte, se cuenta la influencia de los niveles altos de la estructura funcional sobre los bajos.

El tratamiento de los sistemas autónomos generales desarrollado aquí también permite identificar otros tipos de factores que influyen sobre los procesos de inferencia, aparte de heurísticos relativos a la resolución de problemas: restricciones de tiempo real, recursos, coordinación, etc., y factores implícitos derivados del *acoplamiento de substrato*.

Se puede considerar que este trabajo sigue las ideas principales de la percepción abductiva. Las similaridades con [Sha05] y con nociones y puntos de vista recogidos en otros trabajos son claras [Roc85], [Roc97], [Roc83].

Sin embargo, se ha formulado teniendo en cuenta un contexto más general que incluye aspectos sistémicos. Esto permite evaluar su influencia y relevancia en la percepción, y alcanzar un mayor nivel de generalidad.

2. **Percepción directa, *sense-data* y mediada.** Como ya se mencionó, la percepción entendida desde los puntos de vista de *affordances* y *sense-data* son análogos en cuanto a que asumen cierto carácter directo en la percepción. Existen evidencias de ello en algunos contextos concretos que se pueden consultar en la correspondiente bibliografía. Sin embargo, entendemos que (1) pueden explicar aspectos concretos de la percepción, pero que no son generalizables (2) de acuerdo con esto, su alcance se puede establecer en los términos de este trabajo. Desarrollaremos este punto como base para más comentarios.

   - Una primera manera de representar la percepción directa en los términos de este trabajo puede verse en la figura 15.2, p.249. Nuestra noción de *referente* aparece de forma implícita en la percepción directa. Sin embargo, coincide con el observador.[3] Puede observarse que, de acuerdo a esta teoría, el proceso perceptivo consiste de una única fase desde el *estímulo cercano* a la percepción de *affordances*: superficies como soporte potencial [Gib87, p.127], substancias como nutrientes [Gib87, p.128], etc.

     De acuerdo a la percepción directa, los sistemas sensoriales de los animales están adaptados, intrínsecamente, a percibir *affordances*. Es precisamente la razón por la cual la percepción sería directa. Quiere decir que las *affordances* se perciben únicamente a través de *procesamiento de información cercana*. En términos del presente trabajo, eso equivale a decir que la fase de *procesamiento de información cognitiva* no tiene relevancia. Expresaremos esto diciendo que es un proceso *unitario*, i.e.: que da un resultado idéntico a su entrada. Entonces, la percepción directa se puede representar como el caso (a) de la figura 15.3, p.249.

[3] Obsérvese que: (1) El sistema percibe *affordances* en su entorno. (2) las *affordances* "tienen que ser evaluadas *relativamente al animal*" [Gib87, p.127]. En conclusión, el *animal* i.e.: el observador, es el referente del proceso perceptivo.

Podemos observar que esto equivale a decir que la percepción directa tiene lugar cuando se cumplen al menos dos condiciones particulares respecto al caso general desarrollado en este trabajo:

  - El *procesamiento de información cognitiva* es un proceso unitario. Es decir, que el referente representado coincide con las singularidades consideradas por el perceptor.
  - Que el conjunto de singularidades que resultan del *procesamiento de información cercana* tienen significado respecto a la utilidad del entorno. Esto implica que los recursos del sistema están adaptados al propósito: sistema sensorial, relaciones entre cantidades, el resto de procesos en el sistema, etc.

- En cuanto a la percepción *sense-data*, asumiremos que se trata de un concepto más general que el de la percepción directa. Los principales puntos de diferencia entre las dos son:

  1. La teoría *sense-data* admite que los *sense-data* son dependientes de la mente que percibe, mientras que la percepción directa asume que las *affordances* son exclusivamente dependientes del entorno.[4]
  2. La teoría *sense-data* admite que el significado de los *sense-data* puede no estar referido al sistema, sino a las propiedades intrínsecas de los objetos del entorno: un tomate rojo siendo rojo, una naranja siendo redonda, etc.

  Estas diferencias hacen que la teoría de la percepción *sense-data* sea más general, puesto que explica mayores niveles de abstracción —propiedades intrínsecas del entorno, independientes del observador—, y permite explicar la influencia del observador sobre su propia percepción: experiencias pasadas, memoria, etc.

  La percepción *sense-data* puede expresarse en términos del presente trabajo como el caso (b) de la figura 15.3, p.249. Debe observarse que la fase de *procesamiento de información cognitiva* sigue siendo unitario, pero que los referentes del proceso pueden no estar referidos al sistema, a diferencia del caso anterior.

De acuerdo con estas consideraciones, podemos mencionar los siguientes puntos:

1. Ambas teorías asumen una fase unitaria de *procesamiento de información cognitiva*. Esto implica que necesariamente la fase de *procesamiento de información cercana* tiene que estar adaptada a los referentes del proceso. Es decir, que el sistema sensorial debe ser específico para esos referentes: los recursos en los que está corporizado y las singularidades que considera.

   El rango de referentes que se puede percibir está limitado por el grado de especificidad de los sistemas sensoriales. Si un sistema sensorial fuese muy específico, referentes nuevos o modificados no podrían ser percibidos.

[4]A pesar de que, como hemos visto, deben estar referidas al sistema.

2. Ambas teorías se concentran en las características físicas del entorno. La percepción de referentes abstractos, basados en singularidades abstractas o conceptuales no se explica.

   La percepción directa clasificaría este tipo de percepción como *de segunda mano* o convencional [Gib65]. Sin embargo, hay evidencia de que las percepciones *de primera mano* y *de segunda mano* están relacionadas, y que se influencian mutuamente. También que la percepción *de segunda mano* tiene efectos fisiológicos y produce reacciones cerebrales que, en muchos casos, son indistinguibles de las *de primera mano*. Este tema se ha mencionado en diversas ocasiones en este texto.

3. Las *affordances* tal y como se definen en la percepción directa [Gib66], [Gib87] se refieren fundamentalmente a aspectos como *soporte* y *nutrientes*, que en última instancia responden a la *supervivencia* del sistema. En los términos de este trabajo, sería considerada como un *objetivo raíz*.

   Sin embargo: (1) un sistema, en el caso general, podría tener más de un objetivo raíz aparte de la supervivencia. (2) No es una condición necesaria que la supervivencia sea un objetivo raíz en todos los sistemas, especialmente en los artificiales.

4. Los *objetivos raíz*, como se desarrolló en la parte dedicada a los sistemas, son los de más alto nivel de abstracción y más largo alcance temporal. Son realizados por toda una estructura de objetivos de menor nivel, que pueden ser significativamente distintos a ellos. Esta estructura está adaptada a horizontes temporales más cortos y niveles de abstracción más bajos, que se corresponden con los requerimientos instantáneos derivados del entorno y las capacidades del sistema. Por tanto, cuanto más alto el grado de autonomía del sistema, más baja debería ser la especificidad de sus componentes, de acuerdo al principio de mínima estructura.

En conclusión, podemos afirmar que las *affordances* son una aproximación más restringida que los *sense-data*. La falta de generalidad de ambas aproximaciones implica que dejen mútliples aspectos de la percepción sin explicar, especialmente los relativos a los objetos percibidos.

El carácter directo que asumen las dos teorías no permiten explicar la coordinación ni otros tipos de relaciones de dependencia entre la percepción y otros procesos en sistemas complejos, en los que muchos procesos pueden estar teniendo lugar simultáneamente.

Sin embargo debe observarse también, en contra de las aproximaciones puramente simbolistas, que en el caso general debe contemplarse una fase de *procesamiento de información cercana*, si bien deba entenderse que dependerá de la operación del resto del sistema. De hecho, esta fase puede alcanzar altos niveles de desarrollo, incluyendo procesos de ecualización y equivalencia de singularidades muy evolucionados.

3. Percepción Gestalt. El presente trabajo tiene relación con esta teoría en aspectos fundamentales:

Las *singularidades* son en realidad relaciones entre los valores de las cantidades del sistema sensorial. Además, las relaciones de *equivalencia cognitiva*, ϵ, son relaciones entre singularidades.

Por tanto, al igual que la teoría Gestalt, esta aproximación asume que la percepción está basada en las relaciones entre partes.

- La escuela Gestalt asume que la percepción se concentra en el análisis de relaciones como la simetría. Sin embargo, este trabajo no impone ninguna restricción al tipo de relaciones en que un proceso perceptivo puede considerar como singularidades. De una revisión bibliográfica sobre la percepción de bajo nivel se desprende suficiente evidencia sobre la naturaleza heterogénea de las singularidades como para no poder asumir ninguna restricción sobre ellas.

  Ejemplos de singularidades en las que se basan algunos procesos biológicos de percepción son: proximidad/continuidad/simetría de valores en reconocimiento de objetos, proximidad/continuidad de valores en el tiempo para seguimiento de objetos, discontinuidad de valores en el tiempo para cambios del foco de atención, patrones en el espectro de frecuencias para el reconocimiento de voz. Estos ejemplos muestran diferencias intrínsecas entre los tipos de singularidades en que puede basarse la percepción.
- Este trabajo explica la percepción en el contexto más amplio de *nodo*, que a su vez se enmarca en el de *estructura funcional*. Esto implica que la percepción debe estar sometida a más criterios aparte de la solución *óptima*, en contra de la tesis de la Gestalt. Por ejemplo: restricciones de tiempo real, de coordinación, de recursos y la finalidad. La existencia de estas restricciones explica la razón de la diversidad que pueden presentar las singularidades, y también el hecho de que su interpretación no se corresponda con el óptimo en sistemas perceptivos reales.

4. **Teoría de visión de Marr.** Las ideas de este trabajo presentan múltiples puntos en común con la teoría de Marr:

   - Ambas aproximaciones tienen en cuenta la dualidad entre representación y procesamiento. En este trabajo, la dualidad se encuentra en el papel de los *referentes* y en el de la *percepción corporizada* o *implícita*.

     Los referentes determinan en gran medida el *punto de vista* de un proceso perceptivo, y por tanto ejercen influencia sobre las fases intermedias, incluyendo la percepción implícita derivada. La percepción implícita representa la influencia de un proceso perceptivo sobre el resto del sistema. Los referentes también influyen sobre los *objetos percibidos* en el proceso, y por tanto, sobre los demás procesos cognitivos.
   - Ambas aproximaciones distinguen una diferencia cualitativa entre el análisis del *primal sketch* (*estímulo cercano*) y el resto de procesos perceptivos. Sin embargo, desde el punto de vista de este trabajo, no existe una diferencia cualitativa y fundamental entre los modelos $2\frac{1}{2}$-D y 3-D.

De acuerdo a Marr, los modelos $2\frac{1}{2}$-D y 3-D difieren en su punto de vista. El primero está referido al sistema y el segundo es objetivo. Este trabajo asume que cada proceso perceptivo tiene su propio *punto de vista*. El aspecto fundamental que determina el punto de vista es el referente del proceso. Por tanto, puntos de vista centrados en el sistema o neutrales difieren básicamente en sus referentes, pero no existe una diferencia cualitativa entre ellos.

- Existe otro punto en común con la teoría de Marr: la distinción entre los niveles *computacional*, *algorítmico* y *de implementación* en visión. Estos niveles corresponden conceptualmente a los niveles funcionales de los sistemas: *funcional* (nivel de nodo), *algoritmo* y *función corporizada*. De hecho, se desarrolló la importancia de estos niveles respecto a la adaptatividad del sistema al introducir el concepto de *descomposición funcional*.

3.4 Principales novedades de este trabajo

Podemos concluir de los comentarios anteriores que las principales novedades de este trabajo derivan del nivel de generalidad de su perspectiva.

Normalmente otras aproximaciones a la percepción parten de una disciplina concreta, concentrada en una parte específica del problema. Alcanzar generalidad en esas circunstancias implica un proceso progresivo, en fases. Esto explica por qué cada aproximación ha identificado partes aisladas.

Este estudio ha partido del punto opuesto, desarrollando primero una noción de sistema: finalidad, objetivos, organización y comportamiento, integrados en una visión unificada, y complementados con una visión de la operación del sistema: estructura de nodos, elementos de los nodos. Todo ello constituye un contexto para la percepción y permite:

- Establecer una visión integral de la percepción: (1) forma: procesamiento mútliple y distribuido y (2) contexto operacional y de substrato: restricciones de coordinación, comunicación asignación de recursos, dependencia del substrato.

- Identificar implicaciones múltiples, incluyendo, por ejemplo: puntos de influencia del sistema sobre la percepción y vice-versa, procesos no-ideales dentro de la percepción (por ejemplo reconocimiento según criterios no-ideales, en contra de la aproximación Gestalt), la percepción en el tiempo.

- Construir una serie de fenómenos perceptivos que debe ser explicada: el papel de la memoria, la influencia del resto del sistema en la percepción, heurísticos, emociones, etc. Otras teorías de percepción se concentran solo en algunos.

En conclusión, el nivel de generalidad adoptado permite identificar una larga serie de implicaciones, procesos y fenómenos relacionados con la percepción. También muestra —cualitativamente en este trabajo— su importancia relativa entre ellos mismos y respecto al sistema.

English Version

A Foundation for Perception in Autonomous Systems

Chapter 4

Introduction and Objectives

In accordance with the title of this work, the main objective is to build a conceptual foundation for perception in autonomous systems. This derived in two major goals:

Generality: Explaining perception from a general point of view, establishing a common ontology for artificial and biological systems.

Obtaining concepts, principles and relations applicable to artificial system design and to artificial perceptive system design. This objective included a formalization of the ontology.

These objectives were stated under the belief that the levels of complexity and the nature of the tasks needed by current artificial systems exceeded the level of performance enabled by conventional engineering. Generality would eventually permit applying efficient bioinspired solutions to currently unsolved technical problems.

Bioinspiration has existed in engineering since ancient times; perhaps the most known example is the study of the wings of birds in order to build flying artefacts. More recently and related to this work, specific approaches have led to biologically inspired cognitive architectures, of which RCS [Alb99], [Alb95], [GMP+01] and SOAR [New90], [RLN93], [LBCC99] are perhaps the best known and inspiring.

This work falls within a line of research which aims at general principles and theories more than at specific projects and problems. Its focus is engineering knowledge applicable to any problem regarding autonomous system design. Generality is a necessary condition for this.

The aim for generality is not novel. In the recent history of science, the interest for generality, relations, sets and isomorphisms experienced a progressive rise during the XIX century and perhaps reached is zenith in the mid XX century. The title *General System Theory* [vB69] is regarded as the foundation of the theory centered, precisely, in the study of systems as to their *systemhood*, regardless their circumstantial features: i.e. General Systems. This way of understanding science and reality, promising as it remained for decades, seemed abandoned at the beginning of this work. In fact, it was not until well advanced in this research that the Theory of General Systems was adopted as the background for the investigation, in the particular formulation of *An Approach to General Systems Theory* [Kli69].

The degree up to which the objectives have been achieved is discussed in chapter 15. However, it is worth advancing that a complete formalization has not been proposed, and that this work provides only a semi-formal discourse.

The methodology designed for this work followed the ideal of the traditional scientific method, conceptualized in figure 15.1, p.246, consisting of three major phases: (1) experimentation, (2) observation and (3) generalization, in which experimentation serves both as the starting point and as the benchmark against which to check generalizations.

Due to the multidisciplinary nature of this work and the aim for generality, it was necessary to restate the ideal in order to make the investigation possible in finite time and with finite resources.

The stage of 'experimentation' was reformulated into a thorough analysis of the experience of the Group[1] in intelligent control system design, and an extensive bibliographic research in this and other fields, trying to cover all the scope from experimental research to abstract theories. Performing experiments on psychology, neuroscience, engineering, geometry and all other disciplines related to this work would have been impossible. Instead, it was decided to carry out a major effort in order to transform the documental corpus into both a source of general principles and concepts, and an experimental benchmark against which to test them.

This interpretation of the scientific method, envisaged at the beginning of this research, proved essential later, when more and more new fields of knowledge were added to that which started the research: perception in biological systems. Some knowledge domains studied have been mentioned explicitly in the text, while others contributed to form the concepts proposed, but have been left implicit: consciousness, geometry, art, algebra among others.

Among the second, we must remark that consciousness, in all the perspectives considered during this research [Anc99], [Baa97], [Den91], [Hol03], [Lyo95], [Tay99], parallel distributed systems and real–time systems [BW97], [Jal94], [MR86], [RM86], [Sch95] and other miscellaneous sources [Fra95], [KD95], [New90], [Ame99] contributed to developing the distributed conception of systems which has been adopted in this work.

The thesis has been structured following a *general to particular* scheme. The topics which are treated first are the contextual, which are follwed by the more specific ones. Firstly, a discourse on systems establishes the context for perception. Then perception is developed. This is followed by an analysis of real systems. The work concludes with a discussion about the major achievements and conclusions and reference material.

The same *general to particular* scheme has been applied within each part, assessing contextual aspects in the first chapters and progressively entering the more specific ones.

Part II is dedicated specifically to systems, in order to offer a general vision of them,

[1] Autonomous Systems Laboratory, ASLab. http://www.aslab.org/

which are the context in which perception takes place. Concepts of distributed systems, general systems and engineering are integrated in a unified notion. This discourse on systems has intended to prepare the reader for a clear, straightforward discourse on perception which, in other case, would have proved excessively interleaved with systemic considerations.

Chapter 5 offers a short study of autonomy, the problems involved in building autonomous systems, and the different ways in which it has been approached. In this light, it offers an overview of artificial systems addressing their different strategies for autonomous behaviour. Finally it explores some fundamental aspects related with autonomy in systems.

Chapter 6 offers an overview of the major theoretical and methodological source of this work: *An Approach to General Systems Theory*, by George J. Klir, [Kli69]. It introduces the main concepts and ideas which will be used throughout the text.

Chapter 7 is the main exposition of this part. It integrates multiple concepts about systems inherited from many sources in a unified vision. The internal aspects about systems such as their structure are related with external ones such as behaviour and autonomy. This chapter develops the systemic framework in which perception will be explained in part III.

Part III is dedicated to perception. It develops the topic in the context given by the system, in the terms introduced in part II, understanding that the reader should conceive the concepts and perceptive processes within the restrictions and dynamics of a systemic context: environment, objectives, resources, distributed functions, perturbances, etc.

Chapter 8 describes the problem of perception from a global, basic perspective in order to identify the major parts and processes to be explained. It then describes a collection of relevant approaches to perception from this perspective, indicating the exact aspects of the problem in which each study is focused.

Chapter 9 states the main points of this thesis schematically.

Chapter 10 takes the discourse on perception from the introduction to the fundamentals of problem of chapter 8 and develops it into a detailed view of the process and its parts.

Chapter 11 develops the informational or cognitive aspects of perception, establishing a relation with the taxonomical analysis of chapter 10.

Part IV analyzes examples of real systems in detail, in order to illustrate the concepts of parts II and III.

Chapter 12 describes an embedded automotive system for detecting losses of attention in the driver. It is analyzed in detail mainly for aspects on perception.

Chapter 13 describes the CONEX system, an example of a complex intelligent control system, a past development of the ASLab Group. It is analyzed both for systemic and perception concepts. **This chapter was contributed by the director of this work, Dr. Ing. Ricardo Sanz**, who actually took part in the CONEX project.

Chapter 14 analyzes the case of a fault-tolerant, massively parallel system for concepts like reconfiguration, adaptivity and functional decomposition, which are not easily found in artificial systems.

Part V includes chapter 15, in which the major achievements of this work are discussed, and the resulting framework compared to existing theories.

Part VI Includes the initial versions of a glossary on general autonomous systems and specialized terms of perception, and the list of bibliographic references.

Finally, it is worth remarking, although it will be outlined throughout the text, that the concepts introduced here are *general*. This means that they must explain the simple and the complex, the particular and the abstract, the natural and the artificial. No real system is known to the author to fully develop the generality of all the concepts proposed in this work.[2] In real systems, some of the aspects mentioned here may appear in such a primitive form as to be only 'degenerated' instances of our concepts. Others may appear to fully develop the notions proposed here.

One of the objectives of the line of research in which this work has emerged is to be able to design systems in which their characteristics are developed up to an arbitrary degree, at the choice of the designer. Of all characteristics, autonomy would be perhaps the most tempting, and initiates the discourse. Mastering the design of parts and structure of systems will eventually enable this. However, we shall deduce from the text that this objective is so ambitious that it may well become a dream. Let this work be a primitive step in the way to this dream.

[2]'Generality' in this sentence refers to the full extent and meaning of a concept.

Part II

Systems

Chapter 5

State of the Art of Autonomous Systems

In the following sections, we shall try to explore how the concept of autonomy is understood in artificial systems, and how different aspects of autonomy emerge from different designs.

Autonomy is a concept which may lead to a variety of interpretations, because one normally understands it as an abstract quality which is difficult to describe formally, and also because there exist many ways for a system to be autonomous. It has often been tried to build a core notion by focusing on its etymology:

> autonomy –literally *control of the self* from the Greek *autos* (self) and *nomos* (a law)– [Bat01, p.118]

The notion of autonomy as *self-control* underlies, in fact, all approaches relative to artificial systems, although different aspects are emphasized:

- Absence of human intervention [HMH04, WJ94, Ken03, HMH04].
- Minimum dependence of the system from its environment [New90, CH00, Col, BOAJ06].
- System cohesion (unity) [CW04, Col, CH00].

Autonomy results from a certain combination of different capacities and characteristics: fault-tolerance, intelligence, knowledge and response time among others. Some systems are designed stressing more some of them than others, and thus, obtaining a *kind* of autonomy closer to one of the senses mentioned above than to the others.

Autonomy in systems may differ in its intensity as well as in its kind. Indeed, absolute absence of human intervention, absolute independence from the environment and cohesion under any circumstance are impossible. The human is always present, by setting a target for the machine, by partially operating it, or by designing the way it will behave when left to its own. Artificial systems are influenced by the environment in any case, for there will always exist an external factor which may eventually affect the system: gravity, electromagnetic fields or temperature for example. Indeed, all systems may be broken apart and made to lose cohesion. Some artificial systems will be more independent from humans or from their environment than others, and they will also be able to maintain their cohesion differently, as a result, autonomy will be displayed in different degrees.

5.1 Systems

In engineering, a system is conceived as a whole whose interaction with the environment and internal operation can be known by analyzing certain aspects about the way in which it is formed, the way in which its parts behave, and the way in which it interacts with everything around. In general, these aspects may be classified as inputs, outputs, and internal variables, which we will call, generically, *quantities* (see figure 5.1).

The inputs of a system affect the system output; it is usually said that the system uses its inputs to calculate its ouputs. *Inputs* of a system are quantities whose values are forced by the environment into the system, while the *outputs* are quantities whose values are forced by the system into the environment. They represent a *coupling* with the environment, for they are intimately related to both system and environment.

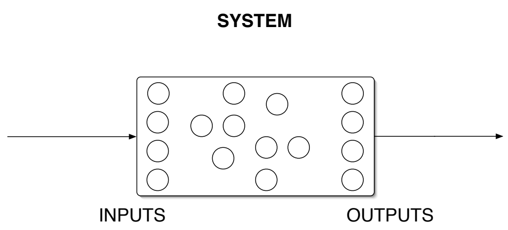


Figure 5.1: Conceptual model of an artificial system. Inputs are represented to the left, outputs to the right and intermediate quantities in between, symbolized as circles.

Artificial systems are designed for a certain task or for achieving a certain goal, or, using a more general expression: artificial systems have defined *objectives*. An objective is a certain combination of values of inputs, outputs and internal quantities of the system. Objectives can also be defined in time, as target values (or ranges) for the quantities to be maintained or achieved within established temporal constraints.

The operation of a system consists in changes on the values of its quantities. Changing inputs of the system may induce changes in other inputs, in internal quantities, in outputs or in all the rest. Analogously occurs with internal quantities and with outputs. These induced changes correspond to certain patterns of influence between quantities which are called *relations*. In most artificial systems, relations may be expressed explicitly as equations or mathematical expressions.

Desgning a system consists in selecting a set of quantities and establishing a set of relations so that the system will achieve the desired objective. In this way, during operation, the values of the system quantities will evolve towards the desired pattern of values. In many cases, the design of the system enables achieving the objective without human intervention, and the system is said to operate *autonomously*.

5.2 Autonomy in Real Systems

Real operating conditions impose restrictions to the ways in which a system can operate autonomously, and also to the types of objectives which an artificial system can achieve. We may classify the difficulties relative to autonomous operation:

Perturbances: Ideally, interaction between system and environment takes place only through its inputs and outputs. The environment influences the system through its inputs and in turn the system influences the environment through its outputs. The system could be designed in such a way that the operator could activate it (figure 5.2 (A)) and then it could continue operating *autonomously* (figure 5.2 (B)), finally achieving its objective.

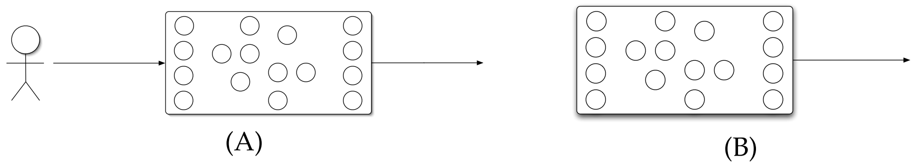
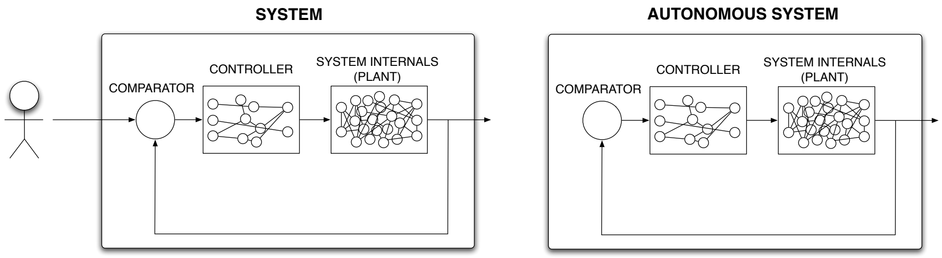


Figure 5.2: Ideal system designed to operate autonomously. Once it had been activated (A), it would continue operating towards its goal without human interaction (B), as a result from the system design.

In real systems, however, this is seldom the case, unless special restrictions are put and the system is let to operate in a *controlled environment*. Although the main interactions between the environment and the system will take place through inputs and outputs in most circumstances, in real operating conditions all system quantities, inputs, outputs and internal variables, may also be coupled to the environment in a lower degree. Eventually, the environment may evolve inducing changes in the system through these copulings. As a result, the ideal engineered system will not represent the real one, which will evolve in a different way, and may diverge from the objectives.

This effect is called *perturbance*, an alteration of the normal operation of the system by the environment. Perturbances are not modelled in the system. In many cases, they cannot be modelled at all, because either their origin is unknown or because they appear randomly. In other cases, modelling perturbances and including them explicitly in the design as inputs would make the system excessively complex or costly.

Classical control theory provided a basic technique to minimize their effect, called *feedback control*. Its principle of functioning is to compare the desired output (objective of the system or *reference*) with the real output the system is exhibiting. If a difference is detected, the system inputs are set in order to correct it. In this way, the effect of perturbances, whichever they are, is avoided. Feedback control is illustrated in the diagrams of figure 5.2.



(A) (B)

Figure 5.3: Feedback control structures in systems, basic mechanism for autonomous artificial systems.

Abstraction: Traditionally, the tasks that have been carried out by artificial systems have involved measurable values, easily formalized mathematically. The basic feedback principle illustrated in figure 5.2 has been refined and modified in multiple ways, optimizing it for different purposes and degrees of performance. If we consider it carefully, we see that the role of the human in feedback systems is left to activating the system and specifying the desired objective by setting a *reference signal*. In many circumstances, the immunity to perturbances achieved by the designer is sufficient to enable long periods of unsupervised operation.

Large systems in the real world, however, are formed by huge quantities of feedback subsystems. Many of them may be independent from each other because they may be used for totally different purposes, but others may function in an integrated way. One of the purposes of integrating separate systems is to accomplish more complex tasks leading to more complex objectives. Think, for example, in optimizing the production rate of a large chemical plant in such a way that it follows the product demand. This objective is regarded as more abstract and complex than manipulating one of the million manifolds in the plant, but both tasks could be, in some way, automated. Indeed, dynamically adjusting the production rate involves many other, simpler actions, so a production rate controller could be said to control other systems. It is a form of controlling control. Incrementing the production rate, will end in opening some valves by changing their reference angles (objectives.) A controller will define other controllers' references as in the figure.

The objective of the production controller is said to be more *abstract* or of a *higher level of abstraction* than that of the manifold controller, which is said to be more *specific*, or of lower level. Different attempts to formalize the concept of level of abstraction have been made, although there is no universal agreement. In industry, generally, higher level of abstraction is understood to refer to tasks and objectives involving longer periods of time. Large industrial control systems are formed, at the lowest level, by servovalves, electric motors and such components, and at the more abstract levels by production planners.

Apart from indicating a correlation with the temporal horizon of actions, abstraction also indicates that an objective or task is based on non-measurable aspects. For example, during the last years there has been a trend to develop artificial systems for interacting in unstructured, human environments. This has introduced human concepts into the design of machines in many ways, for example, building Advanced Driver Assistance Systems (ADAS) that can detect the driver's *fatigue*, or robots with *social* behaviour. Human concepts are generally non-measurable and of a high level of abstraction.

Autonomy in systems operating at high levels of abstraction has several difficulties. First, as an abstract task may involve large quantities of lower-level ones, there exists the problem of finding a suitable way of decomposing it into a lower-level set. In general, there is not one only way and each alternative may realize the abstract task differently. Choosing a particular law of decomposition may have implications on performance and on the system's future actions. There exist two main ways in which artificial systems can perform decomposition: (a) in a fixed way, regardless the evolution of the system, or (b) with mechanisms to adapt decomposition dynamically.

Artificial systems achieving abstract objectives or performing abstract tasks implies that they have to operate internally with abstract concepts. Common artificial systems operating at low levels of abstraction carry out three main kinds of functions: sensing, calculating and actuating. We can see in figure 5.2 how they are associated: the output of the system on its environment is sensed, as well as the reference set by the operator, new inputs are calculated if necessary, and the corresponding actions are taken. The elementary functions of sensing, calculating and acting have to be generalized for abstract systems into perception, deliberation and action.

Designing perceptive, deliberative and complex action functions and systems involves a series of difficulties. In general, abstract concepts to be perceived in the environment may not be sensed directly, but *identified* and *characterized* from particular combinations of the sensor values. Deliberative mechanisms at high levels of abstraction may not correspond to mathematical operations and may adopt more general forms such as those in animal systems, difficult to map directly into artificial implementations. Abstract action has the problem of task decomposition among others.

To summarize, abstraction in artificial systems implies undetermination in all system functions. For system operation, it has to be resolved either as part of the design or dynamically by the system itself. This problem may lead to highly complex processes that may:

- Not have a solution.
- Diverge.
- Require unavailable resources and power.
- Require infinite time.
- Have different solutions (this would require a criterium to choose).

Autonomy requires that the system proves efficient under any of these circumstances.

Uncertainty: While an artificial system may present a clearly defined structure and a reasonably well-defined operation, the environment in which it is going to operate appears as the exact opposite: unstructured and random. In many cases, artificial systems are left to operate within closed rooms, under controlled temperatures or in human-free environments so that the probability of there occurring something unexpected which could affect the system is reduced.

Returning to our conception of system as a set of quantities, we may say that the uncertainty of the environment affects the system by introducing perturbances in any possible way. Consider for example a cruise control device in a car, maintaining the vehicle at a desired speed. The typical perturbance is the arriving to a slope, causing a reduction in the vehicle speed from outside the system (the reference given by the driver, ie: the system input, is not altered.) This type of perturbance is usually compensated by the system. But there eventually may appear an obstacle blocking the car, a perturbance to the cruise control system due to the uncertainty of the environment, which would take it away from its objective, and which the system cannot compensate for.

This example shows the two kinds of uncertainty of the environment, which we could call *intensive* and *qualitative* respectively. Traditional artificial systems, with measurable quantities, usually implement feedback mechanisms that enable correcting deviations in the output induced by the environment, essentially affecting the *intensity* of the controlled variable (output.) Events such as the blocking of the vehicle and the subsequent impossibility to maintain the selected speed are not supposed to be managed by these systems. The occurrence of such events qualitatively change the situation.

Systems operating with abstract quantities, however, are frequently intended to accomplish tasks that may involve both types of uncertainty. Consider, for example, one of the paradigms of the current lines of research in automotion, automatic driving. An automatic car may share many objectives: passenger safety, fuel efficiency and agile transportation among others. While driving, many events may happen due to the uncertainty of the environment, including the appearing of slopes and obstacles as in the previous example. Obstacles may threaten passenger safety, requiring route replanning for avoidance and to ensure arrival to destination. The main aspect introduced by qualitative uncertainty is that it requires the system interpreting its environment, evaluating it with respect to the objectives and reacting to it dynamically, in real time. There are two main ways in which artificial systems may do this: either by choosing among a set of alternative self-configurations included in the system design, or by constructing an appropriate new configuration on-line.

In different ways, all notions of *autonomy* in artificial systems refer to the degree in which a system can deal with perturbances, abstract concepts and uncertainty maintaining its cohesion as system.

5.3 Models of Autonomy

Autonomy is generally understood as an unmeasurable quality of a system, resulting from other characteristics which are also non-measurable such as intelligence, robustness or adaptivity. There have been attempts to reduce this undetermination by building frameworks and specifying related aspects.

Component Models. The actual capacity of an artificial system to compensate for perturbances, and to operate with abstraction and uncertainty is variable in time. This is due, first, to the fact that its own configuration changes in time due to its own evolution and to the influence of the environment (see 'perturbances' and 'coupling' in section 5.2.) Second, the environment itself is in evolution. Finally, the difficulty associated to realizing the objective of the system, due to its complexity and also to the capacity of the system in relation to the environment, also varies in time.

In summary, the autonomy of a system has to be modelled as an evaluation of these three factors: system, environment, task (suggested in [SMG00].) They constitute the *autonomy vector* at a certain instant. In order to build this vector, metrics have to be developed for each the components, which must reflect, on one side, the capacity of the system itself, the influence of the environment and its intrinsic difficulty, and the difficulty of the task. On the other side, they must also account for the relation between the three, for the degree of autonomy depends critically on how system, environment and task are combined.

Developing appropriate metrics is not an easy task. In order to describe the system, for example, it is necessary to choose a set of representative aspects which can be measured, as well as a suitable procedure for calculating the relation between them and the other two components of the autonomy vector. The same happens with the task and the environment. The main difficulties for developing such metrics come from the fact that (a) there is no way of knowing if a relevant aspect has not been counted, and (b) many relevant aspects to be evaluated are non-measurable, forcing to design a numerical equivalence scheme to account for them. All these factors make that the representativeness of a measure of autonomy may be limited in practice, and that an absolute metric (homogeneous for all systems) is therefore difficult to design.

Attempts have been made to identify a collection of the most representative aspects which underlie autonomy, as in *Measuring Performance of Systems with Autonomy: Metrics for Intelligence of Constructed Systems* [Mey00].[1] Some are mentioned following [Mey00, p.18]:

- Long-term planning.
- Various principles of knowledge representation.
- Ability to acquire the data, which characterize and quantitatively measure mission performance.
- Ability to handle sensing, data-processing, and decision making (including planning, navigation, guidance, and control), dealing with uncertainties, especially while operating in the uncertain environment.
- Ability to respond to changes in the environment or its self-state without requiring human intervention.

Application-specific autonomy vectors can be developed by selecting components that are representative of system, task and environment under the specific conditions of a particular application. A metric for autonomy was designed in this way for unmanned vehicles for military applications. The ALFUS (Autonomy Levels for Unmanned Systems) framework considers a three–component vector:

> This model comprises three axes, namely, difficulty of the environment, complexity of the mission, and operator interaction (inversely proportional –less interaction is more autonomous). The autonomy level of a particular UMS [UnManned System] can be represented with a triangular surface with certain values on the three axes. This model suggests vectors, as opposed to single scale, to characterize unmanned system autonomy levels.
>
> [HMH04, section 2]

Qualitative Models. The degree of autonomy in an artificial system can be classified according to the type of functions it can perform. Two main categories are distinguished [GL04, p.2]:

- Decisional autonomy: The capacity of the system changing its own objectives in order to adapt to a particular situation.
- Operational autonomy: Capacity of the system to compensate perturbances when operating towards a given objective.

[1] [BOAJ06] offers a deep insight into autonomy and proposes an entropy-based metrics.

[GL04] offers a five-level scale of decisional autonomy in multi-robot systems based on a taxonomy of decisional functions. The framework is of general application to artificial systems. The classification of functions follows:

1. Supervision and scheduling: Scheduling refers to task management according to fixed rules. Supervised scheduling implies decision taking and dynamic alteration of scheduling rules.
2. Coordination: Integration of the individual actions of system parts [a group of robots] so that they all contribute to the overall objective in an efficient way. This may be implemented at different resolutions, ranging from distributed schemes (negotiation, sensory fusion, cooperation) to centralized schemes (one subsystem [robot] commands the rest.) Prioritization policies, hierarchies and other techniques may be employed.
3. Planning: Designing series of actions directed to achieving an objective. Implies evaluating the relative disposition between the system (and its parts) and the environment. Optimization in terms of system response time, energy efficiency or other aspects may also be considered.
4. Task allocation: This deals with the way to distribute tasks among the different parts of the system (or the different robots of a multi-robot system.) It requires to establish a task assignment protocol in the system, and to define metrics to assess the relevance of assigning given tasks to a particular part.

There are five levels of autonomy depending on which of these functions can be performed by the system [GL04, p.3], as seen in table 5.3.

Level	**C/D**	**Sup/Sched**	**Coord.**	**Planning**	**Task alloc**
1	C	X	X	X	X
	D	-	-	-	-
2	C	x (sup)	X	X	X
	D	x (exec)	-	-	-
3	C	x (sup)	x(high-level)	X	X
	D	x (exec)	x(low-level)	-	-
4	C	-	-	-	X
	D	X	X	X	-
5	C	-	-	-	-
	D	X	X	X	X

Table 5.2: Levels of decisional autonomy in artificial systems depending on the kinds of functions they can perform (adapted from [GL04]). C/D=centralized/distributed system.

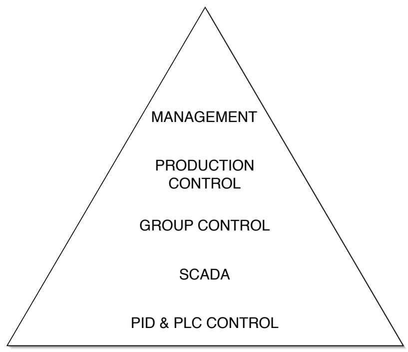


Figure 5.4: Typical hierarchy of control in large systems. At the low level, PID and PLC control is in charge of valve opening, temperature adjustment, etc. SCADA (Supervisory Control and Data Acquisition) systems register data and serve as input to PC-based control of groups of plant subsystems. Global coordination is achieved at production level, and long term strategies (involving marketing, etc.) are taken at the highest level, establishing criteria of operation for lower levels.

5.4 Architectures for Autonomous Systems

Following from the conception of autonomy as *self-control*, we may understand that the autonomy of a system depends, more than its net resources, on its control architecture, that enables it to adjust itself to its environment and its task. The control architecture of an artificial system performs decisional and operational functions to this purpose in different ways. We may informally classify the most common approaches in four categories (for summaries and surveys see [BS01, Fit97, Dav96, WJ94]):

I. Control Architectures. The feedback control topology shown in figure 5.2 is the most widely spread topology in industry, and is an example of an operational control function. In its basic form, control is performed by the PID (Proportional-Integral-Derivative) component, which calculates the appropriate inputs to the controlled part of the system in order to compensate for perturbances affecting the output.

PID control has limitations coming from two aspects: first, the nature and range of perturbances it can compensate for. Second, the nature of the controlled part of the system, that may not be adequate for this type of control (due to non-linear behaviour, for example). Variations have been introduced to the basic control scheme which either dynamically adjust the PID component to adapt it to the situation, or substitute the component with controllers of different kinds (eg: compensators.)

Decisional functions exist in large control systems in industry. In general, these systems are structured following the control pyramid paradigm, by which low level PID controllers are controlled by more evolved systems (PLC: Programmable Logic Controller) and these themselves by software systems implementing planning functions.

In this scheme, PID controllers and PLC devices assume operational functions. Coordination functions are distributed between PLCs and software systems, and task allocation, planning and supervision are carried out by software systems.

Some control keywords and references: nonlinear control, adaptive control, feedback linearization, predictive control, control in state-space, robust control, H_∞ control, SCADA systems, optimal control [Oga90, SL91, Che00, GL00].

II. Reactive and Behaviour-Based Architectures. Control architectures as the above are adequate to operate in environments with limited uncertainty and limited sources of perturbance. Systems designed to move autonomously require avoiding obstacles and determining their position relative to the environment. This implies reacting to a much wider range of situations than those of an industrial environment, with higher uncertainty and more sources of perturbances.

Reactive architectures are designed to make systems act in response to their environment. The output of the system is designed as a function of its inputs, so the action of the system appears as a *reaction* to a certain combination of input values.

Supervision and planning do not exist in reactive architectures. Sequences of tasks are executed as a reaction to the inputs. When the inputs change, on-going tasks are interrupted and replaced by new ones. Coordination may exist in distributed reactive systems such as multi-robot systems, for example, but the coordination rules must be designed as part of the architecture so that coordinated interaction between system elements results as a function of their inputs. Task allocation is also pre-designed in the system architecture and implemented by enable/disable command trees between system elements, which work as a function of the system inputs. As a result, reactive systems may operate within the range of environmental conditions designed in their architecture.

Some reactive architecture keywords and references: Subsumption architecture, new Subsumption architecture, behavior, action-selection [Bro91a, Bro91b, Ark98, BS01].

III. Goal-driven Architectures. This type of architectures are designed to operate in uncertain environments. Typically, uncertainty may be due to the system not having an internal model of a particular environment or situation. Goal–driven architectures are designed to achieve their objectives in these circumstances by designing appropriate actions.

The basic principle of functioning of this type of architectures may be expressed as a cyclic sequence of processes:

- Build an objective.
- Analyze the environment.

- Design a task to achieve the objective within the given environment.
- If this cannot be done, build a sequence of lower level objectives which will realize the higher level one.

These processes may imply highly developed afferent, efferent and deliberative functions. In parallel to them, these architectures may implement learning algorithms that help optimize the system for future or eventual conditions of operation.

These architectures may carry out all families of decisional functions, and thus may achieve high levels of autonomy. However, some limitations have been met when implementing them in actual systems, relative to resource consumption. Some deliberative, learning and perceptive processes would require extremely large memory and computational resources for real time operation, especially in fast-evolving environments, or when dealing with highly abstract tasks.

Goal-driven architectures keywords and references: agent, objective, task, working memory, short/long–term memory, impasse, blackboard, node, hierarchy, [New90, LBCC99, HR95, WB94, Alb95, GMP+01].

IV. Hybrid architectures. Some approaches have integrated reactive architectures with deliberative features of goal-driven architectures in hybrid topologies, in order to increase the overall system efficiency. The usual topology consists on actually separating operational from decisional functions by an intermediate set of task-dispatching functions, which discriminate when a task has decisional content or when it is exclusively operational. The resulting overall architecture is three-layered in: deliberative layer, sequencer and reactive layer.

The reactive layer is formed by modules of behaviour which are interconnected. A particular combination of inputs to the system triggers a specific connection which activates the appropriate modules so that the system executes a task that achieves the objective. The deliberative layer finds new combinations of modules when the efficiency of an existing one may be improved or when unknown inputs are detected. When a successful sequence of behaviours is constructed it is wired in the reactive layer so that it becomes *automatic*.

Hybrid architectures keywords and references: sequencer, deliberative layer, reactive layer, task automation, ATLANTIS [Gat92, Gat99, Gat98, Ore04, DBK91, Sut90, MA05], summary of architectures in [Bry01, Kös00].

5.5 Operational Aspects of System Autonomy

The general principle for autonomy in artificial systems is *adaptivity*. This enables systems to change their own configuration and way of operating in order to compensate for perturbances and the effects of the uncertainty of the environment, while preserving convergence to their objectives. A series of aspects are studied in artificial systems in order to enhance adaptivity:

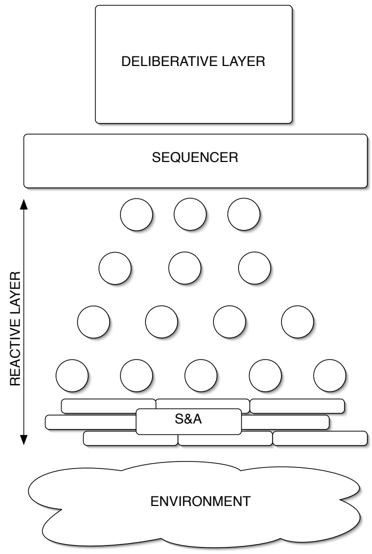


Figure 5.5: Typical configuration of a hybrid architecture. Sensors and actuators of the system (S&A) are employed by the reactive layer, formed by modules of behaviour (circles) operating in sequences as response to system inputs. New sequences for unknown situations (inputs) are designed by the deliberative layer. The sequencer governs the interaction between deliberative and reactive layers.

Cognition: In general, systems which are tightly grounded to the physical substrate have reduced adaptivity, due to mechanical constraints, than systems with cognitive capacities (reflections on cognition and autonomy in systems can be found in [Hey90, CH00, Mey00, Cha04].)

It is understood that cognitive capacities in a system result from a combination of lower level aspects which have been studied in attempts to apply biological principles to artificial systems (eg. [New90, HR95, Alb99, Alb91]):

- Knowledge: Representation, retrieval, ontologies, types (procedural/declarative...)
- Perception: Sensation, interpretation.
- Learning: Automation of tasks, chunking, self-reflection, inference.
- Intelligence: Inference, generalization, particularization, association of concepts.

Modularity: Large systems may result in high levels of complexity and interdependence among parts. In order to structure interaction among system parts, systems may be designed as a combination of *modules*. A module is a part of a system which performs a specific function, and interacts with the rest of the system through a well-defined interface of inputs and outputs. Substituting a module for another with the same function and interface should result in an equivalent system.

Modular systems consist of a structure of parts which interact through their interfaces, presenting an explicit structure and functional decomposition. Interfaces make that dependencies between one module and the rest of the system are determined, allowing interchangeability of modules, as mentioned earlier.

Having an explicit structure and defined dependencies are critical factors for adaptivity. Uncertainty, perturbances and planning may eventually require reconfiguration of system parts, or in the way they interact with each other. Several examples can illustrate this point. First, the mentioned hybrid architectures are based on a deliberative layer reconfiguring behaviour modules of the reactive layer in order to react to an unknown situation. Second, implementing fault-tolerance mechanisms in systems involves identifying sources of error, faulty parts and eventually their isolation or reconfiguration.

Fault-tolerance [Jal94]: System adaptivity depends on its capacity to achieve its objectives under perturbances and uncertainty. Eventually, parts of the system may be damaged or malfunction during operation, compromising system cohesion and therefore its capacity to achieve objectives. Fault tolerance techniques have been developed to provide the system with mechanisms to react to these circumstances by adapting itself. Fault tolerant systems evaluate self–performance in terms of *dependability*, which stands for: reliability, availability, safety and security, of which the first and second are the most significant.

Three concepts distinguished in relation with reliability: a *failure* is a deviation of the system behaviour from the specifications. An *error* is the part of the system which leads to that failure. Finally, a *fault* is the cause of an error.

Fault-tolerance in artificial systems is usually implemented in four phases:

1. Error detection: The presence of a fault is deduced by detecting an error in the state of a subsystem.
2. Damage confinement and assessment: The damage caused by a fault is evaluated and delimited (affected parts are identified and effect on objectives estimated.)
3. Error recovery: Correction of the error to avoid its propagation.
4. Fault treatment and continued service: Faulty parts of the system are deactivated or reconfigured and the system continues operation.

Fault tolerance in artificial systems usually distinguishes between hardware and software. Hardware fault tolerance is based on fault and error models which permit identifying faults by the appearance of their effects at higher layers in the system (software layers.) Hardware fault tolerance can be implemented by several techniques, the most known are: TMR–Triple Modular Redundancy (three hardware clones operate in parallel and vote for a solution,) dynamic redundancy (spare, redundant components to be used if the normal one fails,) and coding (including check-bits to test correct operation.)

Software fault tolerance can be based on a *physical model* of the system, which describes the actual subsystems and their connections, or on a *logical model*, which describes the system from the point of view of processing. In general, software fault tolerance is based on the following fault classification:

1. Crash fault: Fault causes component to halt or to lose its internal state.
2. Omission fault: Causes the component to not respond to certain inputs.
3. Timing/Performance fault: The response of the component is too early or too late.
4. Byzantine fault: Arbitrary fault causing arbitrary behaviour of the component.

Soft computing: In relation with artificial intelligence, a series of techniques have been developed in order to make systems capable of operating with uncertain, imprecise or partially representative measurements. The most relevant techniques are:

- Neural networks.
- Fuzzy logic.
- Expert systems.

Chapter 6

General Systems Theory

The notion of system is common among scientific disciplines as a key concept for modelling different kinds of phenomena referred to sets of matter, devices, components or entities in general. It is not unusual, however, to see the notion of system also in other domains such as economics or sociology. The Theory of General Systems is born under the belief that there exists a common understanding under all specific notions of system which is general to all, and which therefore will be referred to as General System.

6.1 Historical Background of General Systems Theory

Historically there have been different approaches to general systems from different backgrounds, adopting a variety of perspectives and discourses. Ludwig von Bertalanffy, regarded as the pioneer formultaing the concept of General System as we know it today, pointed out incipient symptoms of the existence of such notion back in the 1600s, and tracked its development in the works of known personalities through history (Leonardo, Descartes, Poincaré. . .) up to today. His work *General System Theory* [vB69] represents the starting point of the Theory of General Systems. Indeed, the interest on the general, systemic aspects about phenomena already known and studied had grown significantly during the first half of the twentieth century giving rise to new sicentific perspectives such as the Gestalt theory ([Köh69], [Köh59], [Ell97], [WD04]), whose philosophy is best expressed by:

> The basic thesis of gestalt theory might be formulated thus: there are contexts in which what is happening in the whole cannot be deduced from the characteristics of the separate pieces, but conversely; what happens to a part of the whole is, in clearcut cases, determined by the laws of the inner structure of its whole.[1]

[1]Max Wertheimer, Gestalt theory. Social Research, 11 (translation of lecture at the Kant Society, Berlin, 1924). *From:*
http://www.gestalttheory.net/

The number and diversity of contributors and mind-inspirers is uncountable, many of them reputed scientists as Zadeh [Zad65] or Shannon [Sha48], some proposing unprecedented scientific postulates from perspectives rooted in deep philosophical backgrounds, which sometimes gave rise to intense debate and scientific discussion. A commented state of the art (1978) is offered in [Gai78], and more recent historical perspectives can be found in [Boj04] and [Web]. This work has been inspired in a particular text in the field, *An Approach to General Systems Theory*, by George J. Klir ([Kli69]), representative of a trend of precise formulation (historically opposed to that headed by Zadeh, that defended a less precise conception and gave birth to fuzzy logic).

In order to get to the fundamental concept, let us briefly look at ourselves in our research as if we were ourselves subject of our investigation. It may be as a consequence of custom, we tend to think about systems as something real, to some extent corresponding to a part of the universe, as if our identifying a system were the logic consequence of it really existing outside our minds. This impression, however clear, must be regarded illusory in benefit of thinking that a system is only an instrument of our minds for partitioning an undivided reality, too wide for us to conceive at once. As it will be developed later, the notion of system is dependent on the observer who conceives it, in such a way that the same portion of universe could eventually lead to different systems if studied by different researchers, at different times or for different purposes. This apparent undetermination of any system however, does not make it entirely random or impossible to study systems; the problem is in a way, only a matter of making relativity part of our observations. Invariably, there are aspects of a system which will always occur independently of the observer, and even of the system itself. There will be rules that any system will follow, and there will be patterns of evolution that will share the same causes, however different the systems might be.

6.2 Basic Notions

Let us think about what we understand by system, by considering it in relation to what surrounds it. If all possible entities form the *universe*, a *system* can be regarded as a part of it, which is considered isolated from the rest for its investigation. All which is not system is called *environment*. The different disciplines of science share this general understanding in particular ways, usually differentiated from each other in the criteria for separating the system from the universe.

The observer selects a system according to a set of main features which we shall call *traits*. They will be characterized by the observer through the values of a set of *quantities*. Sometimes, these values may be measured, being the quantities *physical*, such as length or mass. Other times quantities are *abstract*, and they cannot be measured, and their values are observed. The instants of time and the locations in space where quantities are observed constitute the *space-time resolution level*. The values of the quantities over a period of time constitutes the *activity* of the system.

In general, analyzing a system one may find that observed quantities are not sufficient to explain its behaviour. There must exist other quantities, which we shall call *internal*, which play a mediatory part. The observed quantities of the system will be called *external*. We shall call the set formed by all the values of the system quantities at a certain instant the *state* of the system, distinguishing between *internal state* and *external state*.

The main task of the observer is to explain the activity of a system. This will be accomplished by identifying patterns in the activity of the system. The quantities of the system may satisfy *time–invariant* relations, by which the values of some quantities may be expressed as function of others. The set of all time–invariant relations is the formal notion of *behaviour* of the system.

We may realize that the behaviour is due to the *properties* of the system. In other words, a system with different properties would exhibit a different behaviour. The set of all properties will be called the *organization* of the system.

6.3 Kinds of Behaviour and Organization

If we consider a particular system during a particular activity, we may observe that some of the time–invariant relations between its quantities may hold for a certain interval but eventually change. We shall say that these relations correspond to the *local* scope. Observing the same system during a different activity, we may observe that some of the time–invariant relations hold. If we again observe the system during a third activity, we could find that some of these relations would have changed. We would say they are of *relatively permanent*, for they hold for only some of the activities of the system. If we were to observe the system during an infinitely large number of activities, we would find that a particular set of relations would always hold between its quantities. They would be *permanent*. Accordingly, we can distinguish three kinds of behaviour [Kli69, p.43]:

- Permanent behaviour.
- Relatively permanent behaviour.
- Temporary behaviour.

The first may also be called *real behaviour*. The second, *known behaviour*. Temporary behaviour refers to the local scope, for it holds only for sections within a particular activity.

We may observe that permanent and relatively permanent behaviour may not be clearly distinguished from each other when analyzing systems. This is due to the impossibility to test the temporal persistence of relations beyond a restricted range of activities.

Let us return to the organization of the system. We may realize that the different behaviours derive from different kinds of properties. We may distinguish two main kinds, which we shall call *program* and *structure*. The temporary behaviour of a system derives from its program, which is the set of properties of local scope. Permanent and relatively permanent behaviours derive from the structure of the system, which we may in turn classify in *real structure* and *hypothetic structure*, [Kli69, p.44], so that the causal relations are as follows:

$$\begin{aligned} \text{organization} &\longrightarrow \text{behaviour} \\ \\ \text{real structure} &\longrightarrow \text{permanent behaviour} \\ \text{hypothetic structure} &\longrightarrow \text{relatively permanent behaviour} \\ \text{program} &\longrightarrow \text{temporary behaviour} \end{aligned}$$

6.4 Defining Systems

In this section, we are going to present fundamental concepts of systems from two points of view. First, by considering its constant parts. Then, by considering the system from the point of view of its evolution in time. Finally, we shall enumerate the requirements for defining a system.

The study of a system as a whole may result difficult due to complexity or to non-observability of some parts. In order to analyze complex systems, the set of quantities is divided into groups, and each studied separately from the rest, as if it were a system on its own. Generically, each of these groups will be called *subsystem*. A subsystem is also called *element* of the system, to indicate that it is considered a component of it. There may be elements which share a group of quantities. This group is called *coupling* between the elements.

If we conceive the system in terms of its elements, we realize that it is formed by a set of elements, which we shall call *universe of discourse*, and a set of couplings. Elements and couplings are structured following a particular topology which we shall call *structure of universe of discourse and couplings* of the system, and abbreviate by *UC-structure*.

However, the system is not perfectly determined by its UC-structure, for the dynamic aspects of the system are unspecified. In order to complement the description of a system given by its UC-structure, it is necessary to analyze the evolution of the values of its quantities.

If we imagine a system at a certain point of its activity, we will find its quantities at certain values, forming its state. At the next instant of observation, the system will have evolved to a different state. We shall call this evolution a *state transition*. We may assume that, given the system at a certain state, not any transition is possible, or, in other words, that only a set of other states is reachable from the original one.

We may understand that each state is associated to a set of possible transitions. The set of all possible states of the system and their respective transitions form the *state–transition structure* of the system, abbreviated by *SC-structure*.

The necessary information for perfectly defining a system consists of its *primary traits* [Kli69, p.52]:

1. The set of external quantities together with the resolution level.
2. A given activity.
3. Permanent behaviour.
4. Real UC–structure.
5. Real ST–structure.

If a definition contains only some of the five primary traits, it results in a partial definition, that leaves aspects undetermined. In this case, we consider it defines a *class of systems* instead of a system in particular.

Example 6.1 (Quantities, Environment, UC and ST-structures)
Let us imagine we design a simple mechanical oscilator as the one in figure 6.1. When excited, the mass will describe harmonic motion at a frequency of $2\pi\sqrt{\frac{k}{m}}$. This frequency is fixed for constant values of the spring constant, k, and the mass, m, and it can therefore be used as a time reference for a larger system. This principle is used in mechanical watches and clocks.

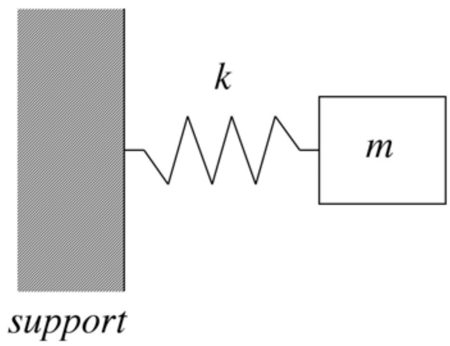


Figure 6.1: Mechanical Oscillator. A mass *m*, coupled to a spring of rigidity constant *k*, coupled to a fixed support.

UC-structure

We may distinguish three elements in the system, which define the *universe of discourse*. They are: mass, spring and support. The couplings between them are as follows:

the mass transmits a force F to the spring. The spring, in turn, fixes the position of the mass, x, relative to the spring's equilibrium point. The spring transmits the force to the support, which returns an equal and opposed reaction force F_R to the spring. On the other hand, the support transmits force F to the environment, which returns a reaction force F_R.

The three elements and their couplings define the *structure of universe of discourse and couplings* of the system (*UC-structure*) shown in figure 6.2.

There is one coupling between system and environment which, for clarity, has not been shown. It is the action of the operator or device (part of the environment) that sets the initial conditions for the system.

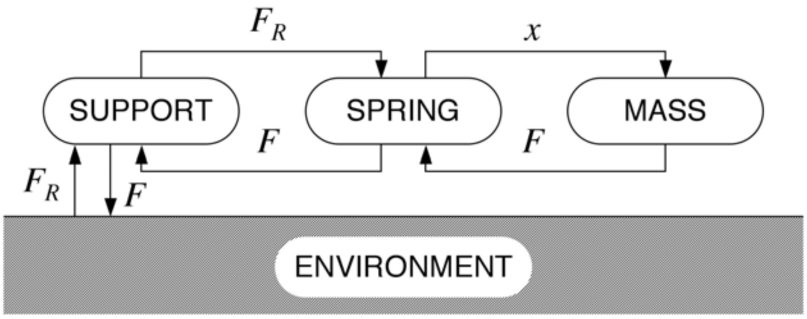


Figure 6.2: Oscillator UC-structure.

ST-structure

In order to analyze the state–transition structure of the system, let us divide operation of the system in three regions, as shown in figure 6.3.

In region 1, the spring admits no further compression, imposing the constraint $x = x_c$. In region 2, the spring follows Hooke's law, and therefore its force is proportional to the displacement from the equilibrium point, x. In region 3, the spring is over its limit of elasticity (at $x = x_t$) and can be assumed as a rigid rod, therefore imposing $x = 0$ and $\ddot{x} = 0$. Although it is not represented in the figure, if $x >> x_t$, the spring would break (region 4.)

These constraints define the states and transitions of the system in regions 1 and 3. Region 2 can be determined by state–space analysis. In this region, the system is described by:

$$m \cdot \ddot{x} + k \cdot x = 0$$

The dynamics of the system is given by this equation and a set of initial conditions.

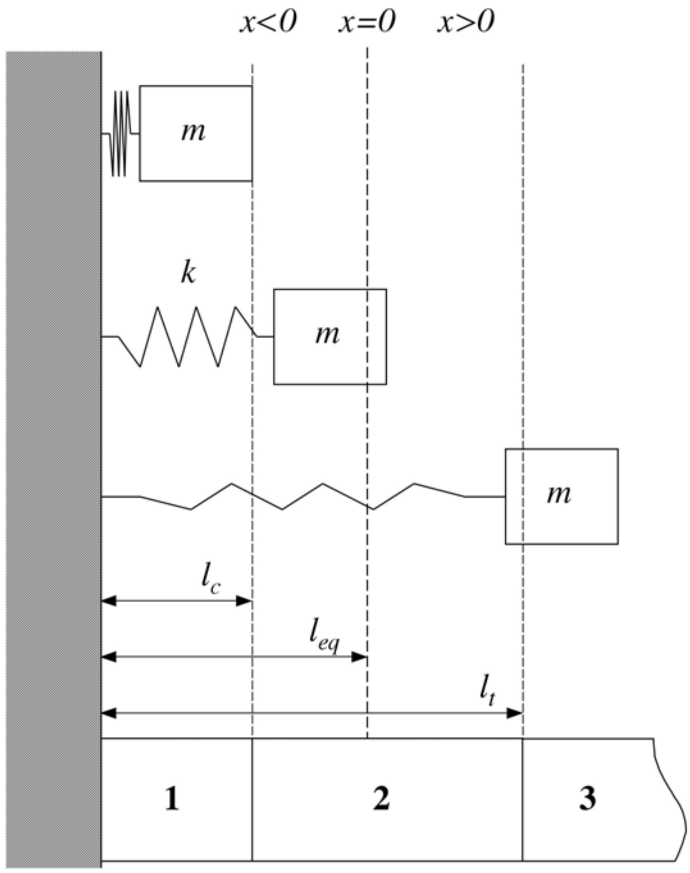


Figure 6.3: Regions of Operation of Oscillator. l_c– length at maximum compression, when the spires of the spring are adjacent to each other. l_{eq}– length at the equilibrium point of the spring, $x = 0$. l_t– length at the limit of elasticity of the spring.

We can consider two state variables, x_1 and x_2, so that[2]:

$$\begin{aligned} x_1 &= x \\ x_2 &= \dot{x_1} \end{aligned}$$

The equation of the system can then be expressed in the classical form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$, where $\mathbf{x}$ is the state vector, A and B are matrices and $\mathbf{u}$ represents the input to the sys-

[2]We might realize that the choosing of state variables is arbitrary. A different x_1 and x_2 could have been chosen leading to a different, but equivalent, analysis. These correspond to the classical analysis of this system.

tem:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

We observe that the system is autonomous, ie: it has no B matrix and no inputs (u).

This system is represented in the phase plane by concentric ellipses (circles if suitable values of k and m are chosen) as shown in figure 6.4.[3] If the mass is set loose at a certain initial position, x_0, the state variables will follow the ellipse containing $x_1 = x_0$.

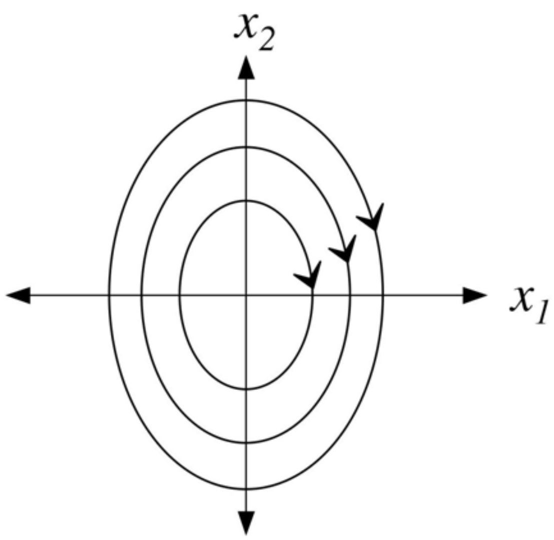


Figure 6.4: Oscillator Phase Portrait in Region 2.

The frequency in which a trajectory is repeated is $f = 2\pi\sqrt{\frac{k}{m}}$, for the solution of the system equation is:

$$x = x_0 \cdot \sin\sqrt{\frac{k}{m}} \cdot t$$

[3]We realize that building phase plane representations (also called phase portrait) of systems might not be straightforward. Tools such as Matlab provide means for this. By hand, two methods are described in [SL91, pp.23-29].

However, this only holds for region 2. Globally, we may understand the phase portrait of the system will be as shown in figure 6.5. The system cannot exist in coloured regions.

To the left of x_c, the spring can be compressed no further. We shall assume that the support will absorb the energy that would push the mass further to the left, to a hypothetical position x_{fc}:[4]

$$\int_{x_c}^{x_{fc}} kx \cdot dx$$

To the right of x_t, the spring is a rigid rod. Any initial conditions x_0, such as points d, are equilibrium points.[5]

In region 2, between x_c and $-x_c$, the system follows Hooke's law and the trajectories are elliptical, as explained above. For initial conditions in $(-x_c, x_t)$, such as points a, b and c, the system follows the corresponding ellipse until the spring can be compressed no further. It then evolves toward the ellipse passing through x_t. This ellipse is, therefore, a *limit cycle*.

Let us consider a set of typical states within the continuum of the figure, as indicated in figure 6.6. The structure of states and transitions for this set is represented in figure 6.7.

As we have mentioned previously, the definition of a particular oscillator is completed by a set of initial conditions. The system portrayed in figures 6.2, 6.6 and 6.7, which stands for many possible initial conditions, stands, therefore, for many particular systems. We can say that these figures define a *class of systems*. In other words, they define a general system, which can exist in multiple, different forms.

In order to use our oscillator in a real mechanical device, we must define a starting point for its oscillation, in other words, a set of initial conditions.

These are the initial values for x_1 and x_2. Physically, initial position and speed of the mass. In figures 6.6 and 6.7, we have portrayed the system under different initial conditions assuming $x_2 = 0$. This is not necessary. For non–zero x_2, the system would follow the corresponding ellipse through (x_{01}, x_{02}). Mechanically, it is more complicated to build such device, and therefore we shall continue assuming $x_2 = 0$.

Let us now consider a particular oscillator, under specific initial conditions, $(x_0, 0)$ so that $x_0 \in (-x_c, x_t)$. Its phase portrait and ST–structure, subsets of figures 6.6 and 6.7, are shown in figure 6.8.

Quantities, State

[4]This is an ideal case. In reality, the energy absorbed by the support, the environment or both would be between 0 and this value. It would be determined by the elasticity of the materials involved.

[5]We have simplified the problem in this region for clarity, by assuming a sudden pass from a spring constant k to a rigid rod. An intermediate region would exist in reality, in which plastic deformations of the spring would occur, by which the system would not recover its position at equilibrium, x_0 (ellipses would progressively shift to the right.) As a result, the dynamics of the system would grow more complex and the phase portrait would show phenomena out of the scope of this text.

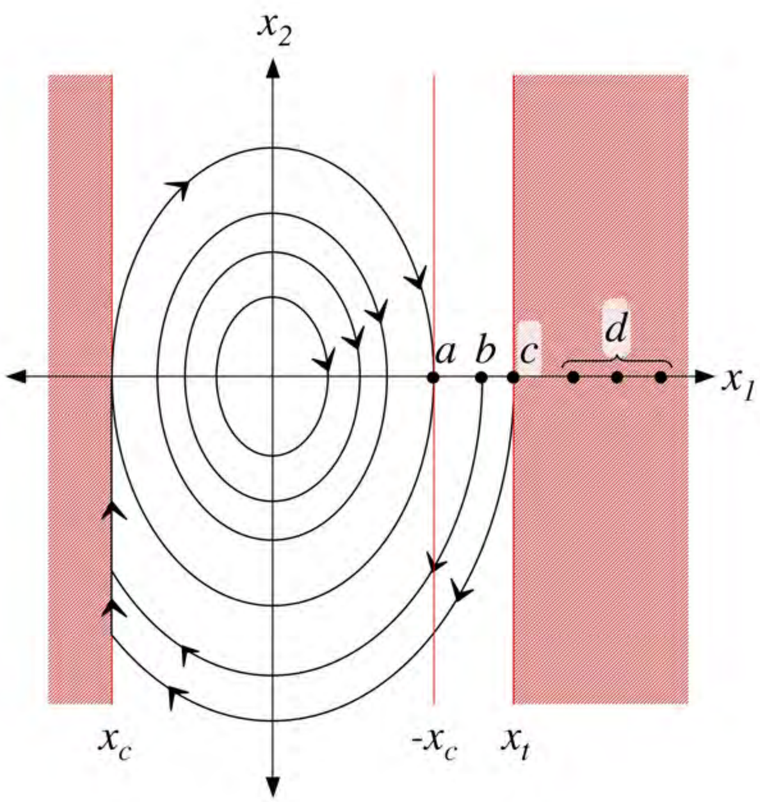


Figure 6.5: Oscillator Phase Portrait.

In order to analyze the ST–structure of the system, we have used two state variables, x_1 and x_2, which have proved advantageous, allowing us to apply powerful methods of system modelling to provide a state–space description of the system. However, we might realize that our definition of *state*, in section 6.2, does not correspond to these chosen state variables. In fact, in our diagram of the structure of universe and couplings, figure 6.2, they do not even appear. Let us see how both views, the (x_1, x_2) on one side, and the (x, F) on the other, come together.

Instead of adopting the point of view of the designer, we shall imagine that we are to analyze an oscillator which is already constructed and working. We are going to imagine that we chose to observe quantity x only (*external quantity*.)

The relation between x and the state variable is straightforward: $x_1 = x$. The exter-

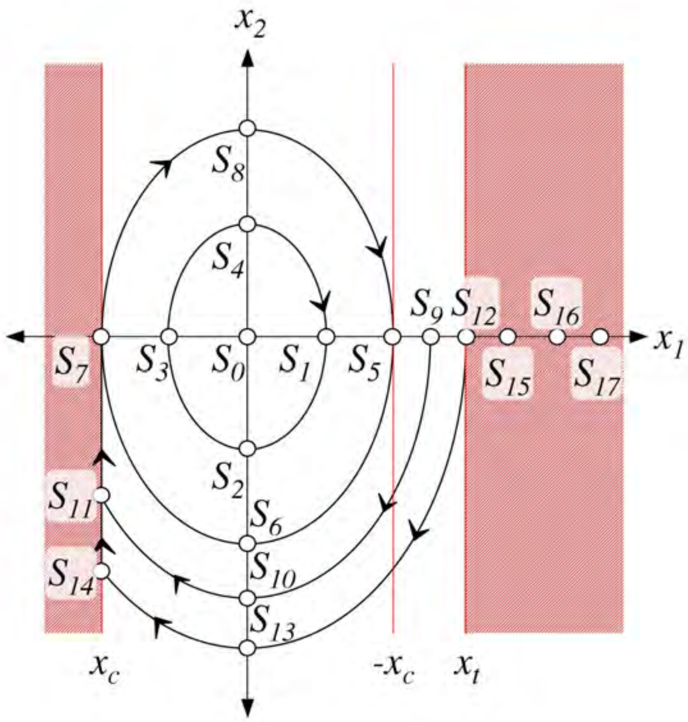


Figure 6.6: Oscillator Typical States.

nal state of the system is therefore equal to x_1.[6]

We should find, however, that the external quantity x would not explain all the aspects of the system. Experimenting with the system, we would find that the part played by k and m would be undetermined. If we stroke the mass during its motion, we would not be able to explain the following values of x.

We could deduce from this that there would exist internal aspects of the system which would remain hidden from out observation. They would disappear if we would consider an *internal quantity* which would reflect in some way the inertia of the mass or its *momentum*. We could well consider the speed of the movement, $\dot{x}$, or its acceleration, $\ddot{x}$. We could then arrive to a set of *time–invariant relations* between its quantities, which would hold in the region of operation of the oscillator:

$$\begin{aligned} m \cdot \ddot{x} + k \cdot x &= 0 \\ x_c < \quad x &< -x_c \end{aligned}$$

[6]We also consider the quantities k, and m, although we shall not mention them explicitly for clarity, understood their values remain constant.

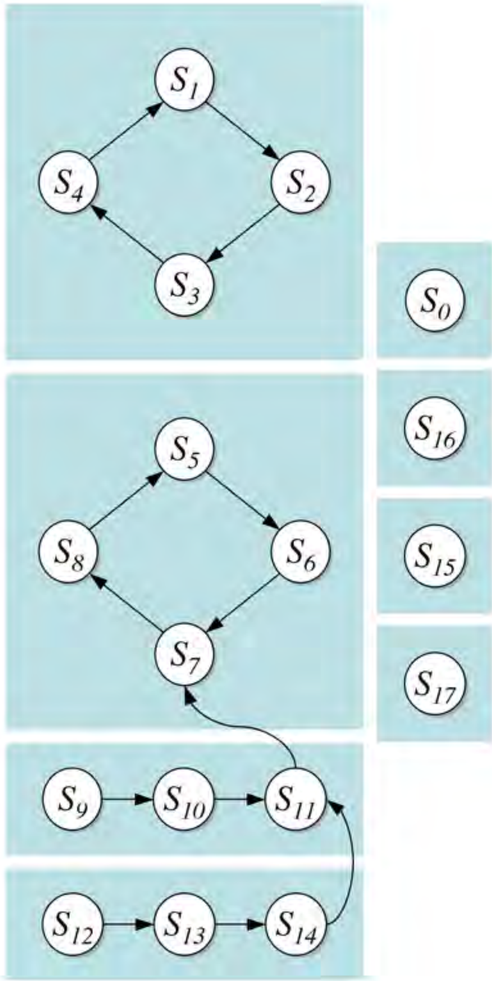


Figure 6.7: Oscillator ST–structure.

In conclusion, the state of the system would be given by (x_1, x_2'), where x_2' would stand for our chosen internal variable. Continuing the analysis from this point, we would arrive to a ST–structure which would be analogous to the above, in terms of x_2. In fact, there would always exist a transformation allowing to represent the system in terms of (x_1, x_2') or (x_1, x_2) indistinctively. ■

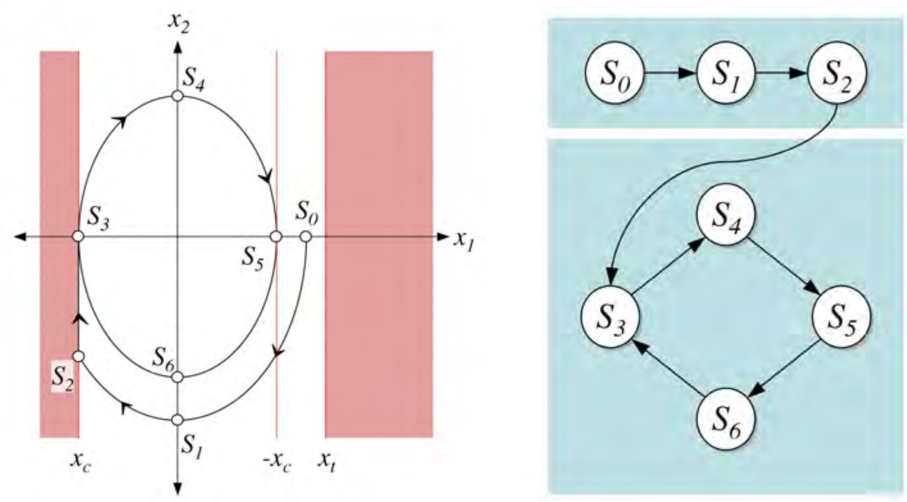


Figure 6.8: ST–structure of a Particular Oscillation.

6.5 Classification of Systems

The concepts of quantity and structure introduced in the previous sections may lead to a classification of systems. We shall consider the short classification of systems illustrated in figure 6.5. The full classification is offered in figure 6.5, taken from [Kli69, p.73].

- Systems
  - physical
    - real → (∗)
    - conceptual
      - bounded → (∗)
      - unbounded → (∗)
  - abstract
    - bounded → (∗)
    - unbounded → (∗)

- (∗)
  - controlled
  - neutral

Figure 6.9: Short classification of systems, adapted from [Kli69, p.73].

Let us briefly explain the categories of systems. We have seen that quantities whose values are measurable are physical quantities, and the rest are abstract. Accordingly, systems formed by physical quantities are physical and the rest are abstract. If we focus on physical systems, we may distinguish two kinds. If quantities really exist, the system is real. If the quantities are only assumed, as in the case of systems which are modelled or imagined, the system is conceptual.

As to the number of quantities and structure a system has, we may distinguish two cases. First, that the system has a finite number of quantities and a finite structure. In this case, it would be a *bounded system*. Otherwise it would be an *unbounded system*. We may see that real physical systems are always bounded, while conceptual or abstract systems may be unbounded.

Finally, if we analyze the quantities of a system, we may find that they can be of two kinds. First, they can adopt values independently from the system, given by the environment. In this case, they are *independent quantities*. Second, their values might depend on the values of other system quantities, and they are called *dependent quantities*. When analyzing real systems, discriminating between dependent and independent quantities is frequently impossible in practice. However, if dependent and independent quantities are known to the observer, the system is a *controlled system*. Otherwise it is a *neutral system*.

6.6 This Approach and GST

In this approach we shall be analyzing autonomous systems and perception from the background of GST introduced in the previous sections. This summary will be enhanced in the aspects required by each topic, introducing further concepts.

The analysis of autonomous system in this text, is written according to several assumptions that define its point of view. Specifically, we shall be considering a bounded, controlled, sequential system as the background. There are several reasons for adopting this perspective.

As to a bounded system, it has been considered because it is the most common case in engineering, and general in cognitive systems. Introducing concepts from the point of view of a controlled system permits explaining them unambiguously. It must be understood that in practice not all quantities and time–invariant relations of a system will be actually known, fact which makes impossible, among other aspects, separating dependent from independent quantities. In many cases, such as normally happens in engineering, the chosen quantities for designing or analyzing a system are few and sufficient to model a system, in which case we may assume the system is controlled. In other cases, the concepts introduced in this text will have to be understood associated to a probability distribution.

Finally, the point of view of a sequential system has been adopted for two reasons. First, because it is regarded as a more general case than a memoryless system. Second, because it is understood that highly autonomous systems are necessarily sequential.

Let us explain the difference between both kinds of system. A memoryless system produces a response which corresponds to the instantaneous stimulus. In some disciplines these systems are called combinational, because past history of either the system

- Systems
  - physical
    - real → (1)
    - conceptual
      - bounded → (1)
      - unbounded → (1)
  - abstract
    - bounded → (1)
    - unbounded → (1)

- (1)
  - continuous → (2)
  - discrete → (2)
  - pulse → (2)
  - hybrid → (2)

- (2)
  - unique
    - controlled → (3)
    - neutral
  - repeated
    - controlled → (3)
    - neutral

- (3)
  - deterministic
    - combinational (memoryless)
    - sequential → (4)
  - probabilistic (stochastic)
    - simple
    - complex → (4)

- (4)
  - anticipatory (teleological)
  - nonanticipatory (physically realizable)

Figure 6.10: Classification of systems taken from [Kli69, p.73].

or the inputs do not influence its output. However, sequential systems are those which use past values of their state and/or inputs to generate a specific output. This implies the existence of a certain memory element to store past values. We understand that a memoryless system is a special case of a sequential system, in which the capacity of the memory element tends to null. Thus, a memoryless system could be analyzed in terms of the concepts of this text by particularizing them for this context.

Systems may only exhibit highly autonomous behaviour if they can react appropriately to uncertainty in their environment. The only means to achieve this is by basing their operation on knowledge. The type and amount of knowledge on one side, and the way in which the system uses it on the other, determine the degree of autonomy of the system.

We would like to add, however, that some aspects of highly developed cognitive systems transcend the point of view stated above. For example, the knowledge of a system may be considered from two perspectives. First, with respect to the resources from which it is formed. Second, relatively to the information represented in it.

In the first case, it can be analyzed as a subsystem, formed by a finite set of quantities more or less related between themselves. These quantities could be, for example, the states (on/off) of the transistors in a RAM memory array. In this case, we can clearly see how the number of quantities is finite and equal to the number of transistors in the memory module.

In the second case, we might realize that the information expressed by the state of the resources (finite, as we have seen,) depends on the way it is interpreted. Returning to the previous example, we could see that the same state of transistors could eventually be interpreted as alphabetical characters, pixel luminance, pixel colours, etc. In general, we may assume that there exists an infinite information, knowledge being, in this sense, an unbounded system. We shall develop these considerations throughout the text.

Chapter 7

Fundamental Concepts of Cognitive Autonomous Systems

We may intuitively conceive an autonomous system as a system that is capable of achieving its objective in an environment. In order to do this, the system may operate either on itself, on the environment or on both. The environment evolves concurrently with the system, and its changes may affect the system. As a result of the mutual influence between system and environment, the way in which the objective is pursued may evolve with time, becoming more direct and fast, slower or eventually divergent from the objective. Effectively autonomous systems have the capacity of maintaining convergence to their objective in spite of the evolution of the environment, and of the system itself.

In this chapter, we are going to explore autonomous systems in order to determine general characteristics of their operation and traits. We shall explore the finality of autonomous systems, the mechanisms of autonomy and general aspects of their architectures.

7.1 Finality

We may understand the concept of *finality* in a system primarily as a certain *objective* to which it directs its behaviour.

An objective is a specification of a state of the system, of the environment or of both. It can be complete, if it specifies all the aspects of the system and the environment, or partial, when it only refers to selected aspects, and the rest are left undefined. A partial objective stands for a class of states. This means that it may equal to a set of states, all of which satisfy the objective specification.

A system that is operating in an environment may actually converge or diverge from its objective. In the first case, we shall say it exhibits *convergent behaviour* or evolution, and *divergent behaviour* in the second.

We shall call *directiveness of the system* to the quality to follow a non–random evolution. In autonomous systems, directiveness represents a trend toward the system objective. The term *finality* is also used in this sense.

Ludwig von Bertalanffy offers a collection of meanings of the term in *General System Theory* [vB69, p.77-80]:

- Static teleology or fitness, meaning that an arrangement seems to be useful for a certain "purpose." Thus a fur coat is fit to keep the body warm, and so are hairs, feathers, or layers of fat in animals. Thorns may protect plants against grazing cattle, or imitative colorations and mimicries may be advantageous to protect animals against enemies.
- Dynamic teleology, meaning a directiveness of processes. Here different phenomena can be distinguished which are often confused:
  - Direction of events towards a final state which can be expressed as if present behaviour were dependent on that final state. Every system which attains a time-independent condition behaves in this way. (*)
  - Directiveness based upon structure, meaning that an arrangement of structures leads the process in such way that a certain result is achieved. This is true, of course, of the function of man-made machines yielding products of performances as desired. In living nature we find a structural order of processes that in its complication widely surpasses all man-made machines. Such order is found from the function of macroscopic organs, such as the eye as a sort of camera and the heart as a pump, to microscopic cell structures responsible for metabolism, secretion, excitability, heredity and so forth. Whilst man-made machines work in such a way as to yield certain products and performances, [. . .] the order of living systems is such as to maintain the system itself. An important part of these processes is represented by homeostasis [. . .]

  . . . equifinality –i.e., the fact that the same final state can be reached from different initial conditions and in different ways. This is found to be the case in open systems, insofar as they attain a steady state. It appears that equifinality is responsible for the primary regulability of organic systems–i.e., for all those regulations which cannot be based upon predetermined structures or mechanisms, but on the contrary, exclude such mechanisms and were regarded therefore as arguments for vitalism.
  - Finally, there is true finality or purposiveness, meaning that the actual behaviour is determined by the foresight of the goal. This is the original Aristotelian concept. It presupposes that the future goal is already present in thought, and directs the present action. True purposiveness is characteristic of human behaviour, and it is

connected with the evolution of symbolism of language and concepts [...] (**)

We may see that (*) and (**) are not equivalent. (*) is formulated from a point of view external to the system. In other words, it stands for an intuition of the observer about the system. The observer expresses the evolution of the system *as if it were* dependent on a final state. In (**), however, the future goal is actually used by the system itself for directing its action.

We shall be using the meanings of 'finality' separately in most cases in this text. However, it is useful to clarify and compile them, for they address major issues about autonomous systems:

1. **Objective:** A desired state for a pair $(system, environment)$ —complete or partial.

2. **Adequacy or fitness:** The capacity of a system for some particular purpose or application. The purpose or application is called the finality of the system.

   If 'finality' is used from the point of view of the system, to refer to some object in the environment, it stands for the potential applicability of the object by the system. It is a generalization of the concept of *affordance* in the ecological theory of vision proposed by Gibson [Gib66, p.127].

3. **Trend:** A trend of the system towards a particular state. It expresses the non–randomness of the evolution of the system.

4. **Structural directiveness:** The patterns of behaviour of the system, understood that they derive from a certain organization.

5. **Equifinality:** The quality of a particular state of being reachable by a system from different initial states and conditions, and by different ways. We shall call *region of equifinality* of a certain objective to the set of possible states of the pair $(system, environment)$ which converge to it by structural directiveness.

6. **Purposiveness or *purposive directiveness*:** Capacity of the system to change its organization, and therefore its behaviour, in order to establish, maintain or improve convergent evolution by explicit consideration of its objective, self and environment.

All these meanings are related to the evolution of the system toward an objective, a major aspect of autonomous systems. In this text we shall be using the specific terms instead of the generic 'finality', except when this is regarded more clarifying.

7.2 Directiveness

Let us consider a semi–formalization of the concept of directiveness as a relation **D**. Denoting a system by S, within its environment, E, and a system objective O. We may consider the expression:

$$(S,E)_1 \xrightarrow{\mathbf{D_O}} (S,E)_2 \quad t_1 < t_2$$

We can read that the system and its environment reach the state $(S,E)_2$ from $(S,E)_1$ following the directiveness of the system, relation $\mathbf{D_O}$. If the system is behaving in convergent evolution, we understand that (S,E) is closer to O in t_2 than in t_1, and that both states are *equifinal* to the desired state O: they belong to the *region of equifinality* of O for system S in environment E.

This formalization of directiveness is generic. In the following sections, we shall analyze structural and purposive directiveness specifically.

7.2.1 Structural Directiveness

As we have mentioned, structural directiveness refers to the behaviour of the system which derives from a particular organization or structure. Let us consider two instants of time, t_1 and t_2, such that the first is earlier than the second. Let us assume that the behaviour of the system may compensate for perturbances during the interval (t_1, t_2). In these circumstances, we may assume that the region of equifinality at t_1 is equal to that in t_2 as for the convergence of the system. Let us denote it by Γ. We have:

$$(S,E)_1 \xrightarrow{\mathbf{D(S,E)}} (S,E)_2 \quad t_1 < t_2$$

$$(S,E)_1, (S,E)_2 \in \Gamma$$

This means that the behaviour of the system, driven by structural directiveness, will drive system and environment from the state at instant t_1, represented by $(S,E)_1$, to a different one, $(S,E)_2$. Both configurations of the system, which we may indicate by S_1 and S_2, would belong to Γ. We might observe that structural directiveness, represented by $\mathbf{D}$, is indicated to depend on the system and the environment. The objective is therefore implicit in the system.

We may understand that structural directiveness implies convergent evolution for a certain region of equifinality. If Γ would change remaining the system unaltered, convergence might not occur.

In fact, in real operating conditions, Γ will hold only for limited intervals.[1] Eventually, the uncertainty of the environment (or the implicit influence of the environment on the system) would cause perturbances which the current organization of the system could not compensate for. These would stand for significant changes in the environment, meaning changes in the region of equifinality, which would become Γ'. In these circumstances, structural directiveness might not drive the system to O. In order to recover convergent evolution, the system must alter its organization to establish a new

[1] As we have seen in chapter 5, variations of Γ constitute a problem for the stability and efficacy of industrial systems. This is usually overcome by building controlled environments to damp the natural changes of Γ.

directiveness within the new region of equifinality.

This is equal to altering the organization at t_1. The extent of reconfiguration needed depends on the differences between Γ and Γ', and on the actual properties of the system. [2]

Assuming that only a part of the system organization is reconfigured, we may realize that it can happen in two ways. First, it may derive from more general levels of organization, in other words, from parts of the organization of longer temporal scope. In this case, the process of reconfiguration would be a result of the arrangement of structures of the system, and therefore, it would be structural directiveness.

But re–organization may not be possible through structural directiveness for various reasons. For example, the structure of the system may not have the capacity of automatic reconfiguration. It may also happen that the actual way in which the reconfiguration must take place is unknown to the system.

In this case, it arises the problem of finding a solution for convergence in the given scenario. This implies explicit modelling of the problem, knowledge, inferential processes, which stand for purposive directiveness.

7.2.2 Purposive Directiveness

Purposive directiveness is the reconfiguration of parts of the organization of the system through processes which operate with an explicit representation of the objective of the system. The symbolic processing determines a new topology for the system organization such that convergent evolution is established, optimized or improved.

Following the notation introduced previously, we may see that purposive directiveness changes the system, S, so that the global state at instant t_2, $(S, E)_2$, falls within the region of equifinality of O. Differently from the case of structural directiveness, the objective is explicit. The process starts at instant t_1, in which the region of equifinality is Γ_1.

$$(S, E)_1 \quad \xrightarrow{\mathbf{D(S, E, O)}} \quad (S, E)_2 \quad t_1 < t_2$$

We may realize that the mechanisms of purposive directiveness of the system operate on the known state, $(S, E)_1$, and a representation of the corresponding region of equifinality, Γ_1. Their operation results in $(S, E)_2$. This state is assumed to fall within the region of equifinality at t_2, which we shall denote by Γ_2.

This is the region expected at t_2 by the mechanisms of purposive directiveness of the system. However, during the interval (t_1, t_2), the region of equifinality might change independently from the system, as a result of the evolution of the environment, becoming $\Gamma'_2 \neq \Gamma_2$.[3] The new state, $(S, E)_2 \in \Gamma_2$, therefore might not be in Γ'_2, leading to divergent behaviour.

[2] In section 7.5 we shall analyze the organizational factors of the system that determine this.

[3] The system itself might also change independently of its mechanisms of purposiveness during this period, also contributing to modify the region of equifinality. Unstable or strongly perturbed systems are examples of this case.

In this light, we may realize that the processes of purposive directiveness of the system may be variably developed, depending on the knowledge of the system. Knowledge equals to more accurate representations of S, E, O and Γ, improved algorithms and metrics for evaluating convergence, and more optimized inference processes for problem solving.

7.3 Objectives

In this section, we shall try to analyze objectives in autonomous systems. We may understand an *objective* as a state of the system, of the environment or of both, to which the system tends as a result of its behaviour.[4]

As we mentioned previously, the state of the system is the value of all its quantities at a particular instant of time. On the other side, the state of the environment represents its situation relative to the system. In other words, it must represent a characterization of the environment according to the parameters which are observed by the system. These are the quantities of the coupling system–environment. The state of the environment relative to the system would therefore equal to the values of the quantities of the coupling. We shall call this notion the *strict state of the environment*.

There exists a slight point to be specified with respect to this. We may assume that the system perception of its environment will not be limited to the quantities of the coupling. Upon them, the system may build developed, conceptual quantities. This makes that, in reality, the state of the environment, from the point of view of the system, will not only consist of the values of the coupling quantities, but also of its conceptual representations of it. We shall call this the *subjective state of the environment*. Unless stated otherwise, we shall understand *state of the environment* in this sense.

An objective is therefore a desired sate of the pair $(system, environment)$. Following the notation introduced in section 7.2, the objective is a special case of a pair:

$$(S, E)^O$$

It must be observed that an objective is *conceptual* because it refers to a desired state, which does not exist in reality.[5] We shall see in the following sections how an objective may appear in the actual, physical operation of the system.

7.3.1 Objectives and Organization

As we mentioned previously, the behaviour of a system will direct its evolution toward an objective. In artificial systems, the objective is set by the designer. In natural systems it results from evolution.

[4]Note that we refer to an objective *of the system*. We shall not refer to the objective of the designer except stated explicitly. The text develops the notion of objective to which the system converges and with which the system may operate.

[5]The objective may specify the desired strict or subjective states, or both. As it was mentioned in section 7.1, the specification can be complete or partial.

The objective constitutes the aspect from which system directiveness is derived. In other words, the characteristic properties of the system, which define its behaviour in the environment, correspond to the objective.

> [behaviour is] a time-invariant relation specified for a set of quantities and a resolution level, and based on samples of a certain pattern (...) If the system exhibits a particular behavior, it must possess (...) certain properties producing the behavior. These properties will be called the *organization* of the system
>
> [Kli69, p.43]

We can understand that the objective determines a specific composition of its properties, leading to a corresponding behaviour. A different objective would lead to different properties and thus, to a different behaviour. We may therefore consider the fundamental relation of causality for autonomous systems:

$$\text{objective} \rightarrow \text{organization} \rightarrow \text{behaviour}$$

This illustrates the conceptual relation between objective, organization and behaviour. In complex systems this relation holds ideally, but there may exist a set of multiple objectives instead of one. Each objective may be related to part of the properties in the organization, in turn leading to different aspects of the overall system behaviour.

In the following sections we shall see how objectives are related to each other and how they define the system finality, structural and purposive directiveness.

7.3.2 Structure of Objectives

Let us analyze in more detail the role of multiple objectives in an autonomous system. We may assume, for the sake of generality, that the set of objectives of a system is heterogeneous. Each of the objectives may differ arbitrarily from the rest. The differences may appear in multiple aspects, although we may categorize them in two main classes:

Time–scope: We shall call time–scope of an objective to the time necessary for the system to realize it.

Level of abstraction: Objectives referred to physical quantities of the system are of a low level of abstraction. Objectives referring exclusively to abstract quantities are of high level. A stronger dependence from abstract quantities will be regarded as higher level.

We may observe that time–scope and level of abstraction are not disjunct nor independent, in that higher level of abstraction will normally be associated to longer time–scope.

Without loss of generality, we may assume that a set of objectives is organized according to a certain hierarchical structure of dependence, which we shall call *objective* *structure*. Objectives of lower level of abstraction and time–scope contribute to realize objectives of higher level and longer scope. We shall use the terms *lower objectives* to refer to the first with respect to the second, which we shall call *higher objectives*.

The hierarchy is formed, on one side, by a group of objectives of the shortest time scope and lowest abstraction. We shall call them *local objectives*. On the other side, by objectives of the longest scope and highest abstraction, which will not contribute to realize any higher objectives. We shall call them *generative objectives* or *root objectives*.

In these terms we can conceive the objective structure metaphorically as a cascade. At the highest level are the root objectives. They *decompose* into a set of lower ones which contribute to their realization. These, in turn, decompose into a lower level, and so forth until the level of local objectives.

Let us make a brief parenthesis in order to mention two classes of dependence between objectives. Each individual objective in the structure equals to a specification of a pair, as in the case of the general notion. If we consider a certain objective i, it will equal to the specification $(S, E)^{Oi}$. If we would analyze the objectives on which it depends, that is, the ones which contribute to its realization, we would find two possible cases. Let us consider objectives j and k, lower to i:

1. O_i may depend *directly* on O_j, if $(S, E)^{Oj} \cap (S, E)^{Oi} \neq \emptyset$.

2. O_i may depend *indirectly* on O_k, if $(S, E)^{Ok} \cap (S, E)^{Oi} = \emptyset$.

In other words, O_i may depend directly of an objective O_j, if they share a part of the specification. It is clear that achieving O_j will partially realize O_i. If the specifications of lower and upper objectives are disjunct, the relation of dependence is indirect. This will generally mean that realizing a lower objective may indirectly contribute to the higher one, by reaching a state from which the higher objective may be realized better.[6]

An autonomous system converges to its root objectives by realizing lower ones, which are simpler or of shorter term. The behaviour of the system tends to progressively realize all the objectives in the structure. It follows the sequence derived from the dependences between objectives. In this way, the objective structure actually defines the trend in the evolution of the system, which constitutes its directiveness.

As we have seen, objectives may have different time scopes and levels of abstraction. This means that some objectives are realized, contributing to achieving higher ones. Eventually, new objectives may also appear. Thus, the objective structure exhibits a certain dynamics. This is a factor for system autonomy. Let us consider several points:

- As long as generative objectives remain unaltered, the rest may be created, eliminated or modified dynamically.

[6] Although the region of equifinality is not being mentioned in the text for the sake of clarity, it is intimately related to objectives and the objective structure. We may realize that realizing objectives may be formulated in terms of directiveness as in section 7.2, and all the considerations about Γ apply. We might add that achieving a particular objective may alter Γ, which is a characteristic to be exploited by the system.

- In this way, the directiveness of the system may adapt to the changing scenario of operation, in order to preserve convergence to the generative objectives.
- Having the generative objectives decomposed into the structure increments tolerance to perturbances. Eventual divergence from lower objectives due to perturbances may affect only parts of the structure.
- Analogously, it is possible to adapt parts of the structure leaving the rest unaltered.
- Having multiple objectives allows separate, dedicated operation which can be carried out concurrently.

Example 7.1 (SOAR Goal-driven Architecture)
The SOAR architecture [New90], [RLN93], [LBCC99] was designed for constructing intelligent systems of general purpose. Different extensions of the architecture have been built to adapt it to specific areas, essentially robotics and aircraft control.

Soar is a goal-driven architecture. The goals are achieved by driving the system through a sequence of states. The knowledge of the system is used to plan the next step towards the goal.

The dynamics of a Soar-based system follows the execution cycle, which consists, essentially, of three steps:

1. Perception, by which the actual situation is characterized. Sensory information characterizes the environment and the system, and this information is matched to the system knowledge. The result is that different possible actions (stored in the knowledge base: production rules) applicable to the actual state are identified. The chosen actions are called operators.
2. Deliberation, by which the different actions are evaluated and a specific one is selected.
3. Action, by which the procedure of action is mapped to the system actuators.

This sequence would eventually lead the system to its goal. In Soar, this goal and the whole set of eventual auxiliary requirements such as would be performance requirements (execution times, precision, etc.), cost functions or safety safeguards constitute the generative set of objectives of the system. However, the execution cycle may not always be executed directly, for special contingencies may occur at any of the three steps.

Let us consider the system at the stage of deliberation, if, for example, the system may not produce a concluding prioritization of the different possible actions to take, and therefore no one can be selected. This is the situation illustrated in goal level 2, figure 7.1. In this case, the system reaches what is called an impasse, implying there is a stop in the progression towards the goal.

In this case, the Soar architecture reacts by generating a subgoal. Its form is equivalent to the system goal in that it is represented with the same structure, but the aim of the subgoal is to resolve the impasse. Once the subgoal is created, the system leaves the original goal aside. The subgoal is considered by the system as if it were its objective, and operates with it in the same way as it would in the original situation: by following an execution cycle directed to achieve it. If another impasse should occur, the process would be repeated, creating a new subgoal aimed at resolving the first.

In the case described by figure 7.1, the application of operator o3.1 would lead to achievement of goal 3. This would permit selecting one of o2.1, o2.2 or o2.3, whose application would resolve the impasse at level 1, eventually achieving the goal.

As a result of this way of functioning, a Soar system at a certain instant of time may be solving a chain of goals and subgoals which correspond to the intermediate objectives, and constitute the objective structure of the system.

Now let us consider in detail the action phase of the execution cycle. This is the phase in which operators are applied. Soar functioning is based on a portrait of the situation of its environment, which in SOAR terminology is called state, which equals to the notion of subjective state introduced in this text.

The system operation consists in changing the state. It may do it in two ways: *directly*, by modifying the portrait. This is called thinking or internal problem solving in SOAR.

The second way in which the system may change the state is *indirectly*. This consists in modifying the environment through its actuators, and subsequently updating the state in memory thorugh new perceptive input.

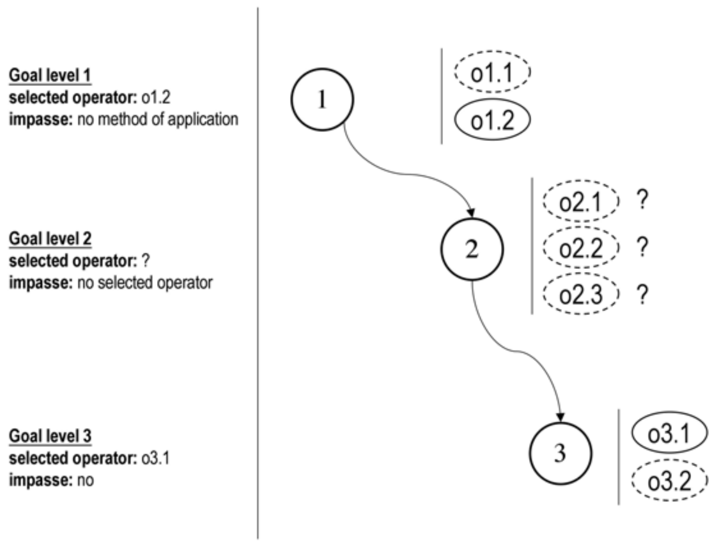


Figure 7.1: Example of a three goal situation in a SOAR system.

In this case, applying an operator is equivalent to mapping it to a series of motor commands and then executing them. The operator may be expressed at a high level of abstraction, motivating a complex pass to the ultimate motor-command level (combination of primitive operators [RN95, p.372].) In artificial intelligence terminology, this mapping is referred to as hierarchical decomposition, operator reduction, operator expansion and hierarchical task network planning [RN95, p.372]. ■

7.3.3 Categories of Objectives

We may realize that root objectives constitute a part of the *definition* of the system itself. In artificial systems they stand for the primary objectives of the designer. They underlie the longest time–scope of operation in the system and they establish the highest level of abstraction. They are a constitutional part of the system, as other fundamental properties, all of which form its *real structure*:

root objective → real structure → permanent behaviour

In accordance with root objectives and real structure, the permanent behaviour represents the basic structural directiveness of the system. It is a set of patterns of action which is intrinsic to the system. In other words, a set of time–invariant relations between quantities, that hold during the life of the system. Therefore, permanent behaviour represents a set of constraints for relatively permanent and temporary behaviour.

As the root objectives, real structure and permanent behaviour are constant in time by definition; we may deduce that the adaptivity of the system relies on the rest of objectives, the hypothetic structure, the program, and correspondingly, the relatively permanent and temporary behaviours. We shall call these objectives *intermediate objectives*. *Local objectives* are the intermediate objectives of shortest scope.

1 (Root, intermediate, local objectives) *In example 7.1, illustrating the major operative aspects of the SOAR architecture, we may observe that the system builds a hierarchy of objectives formed by* goals *of levels 1, 2 and 3. We may consider* goal *of level 1 as a* root objective*, and the rest as* intermediate objectives*.*

Local objectives *are not explicitly shown in the example. We may identify them by thinking about a selected* operator*. A selected operator is a set of actions in sequence. Each of the actions may be elementary, such as achieving certain speed in a motor, or driving an actuator to a certain state. These type of elementary targets are the* local objectives *of the system.*

Intermediate and *local* objectives correspond to the *hypothetic structure* and to the *program* of the system respectively, as the *root* objectives correspond to the *real structure*:

root objectives →	real structure	→ permanent behaviour
intermediate objectives →	hypothetic structure	→ relatively p. behaviour
local objectives →	program	→ temporary behaviour

As we have mentioned, *generative objectives* are realized through *intermediate objectives*, which are of shorter time–scope and complexity. These, in turn, are realized by *local objectives*. This can be clearly observed in example 7.1.

2 (Objective structure) *A basic* objective structure *can be observed in example 7.1 of the SOAR architecture. In this example, an objective is realized by a single objective, and this in turn by another one.*

Objective structures may grow more complicated if an objective could depend on multiple objectives to be realized. In this case, issues proper of parallel systems would arise, such as synchronization, resource–sharing, etc.

Intermediate objectives are defined by substates of the system, the environment or both. This means that an intermediate objective is a desired state of a part of the system or the environment isolatedly, or a relative state between both. In relation to another objective higher in the hierarchy, we may contemplate the two possibilities advanced in section 7.3.2:

- The lower is a substate of the higher objective (direct dependence.)
- The lower is not a substate of the higher (indirect dependence.)

7.3.4 Order

It follows from the discussion about categories of objectives developed in the previous section that the objectives of the system may differ in multiple aspects. Time–scope, abstraction, dependence, precedence, etc. The objective structure expresses this partially. Dependence between objectives is shown explicitly, and this admits certain inference about abstraction levels and time–scopes.

In this section we shall introduce the concept of *order* of an objective. It is a parameter designed for evaluating the relevance of a specific objective within the objective structure. We might realize that objectives in a system may form a highly heterogeneous set, given that they will differ not only in scope and abstraction level, but also in their actual purpose and role.

Because of this, designing a parameter for comparing them on a common basis is not straightforward. It has to be done by considering their most general qualities, such as abstraction level and time–scope. These qualities, in the majority of cases, can only be estimated, making the calculation of *order* a difficult task. In this section we are going to propose a semi–formal conceptualization, and to discuss these aspects in more detail. This will help to identify the exact sources of uncertainty in calculating order, and the representativity that can be expected.

It must be remarked that the *order* of an objective constitutes knowledge about the system at a particular instant. Eventually, a system itself could be designed to use this concept to guide its behaviour, as part of a more complex self–model.

3 (Order as self–model) *A fault–tolerant,* massively parallel system *will be analyzed in chapter 14. In these systems, it is not unusual to have to find a tradeoff between achieving objectives and the actual capacity of the system for doing so (available resources, performance, etc.) This means that some objectives* *eventually have to be discarded in favour of others. Having a concept of* order *would provide a basis for decision regarding the relevance of the system objectives, and help prioritization.*

The behaviour derived from intermediate objectives of high level of abstraction and long time–scope may be similar to permanent behaviour. On the other hand, the behaviour derived from low–level intermediate objectives will be similar to temporary behaviour. In fact, distinguishing between root, intermediate and local objectives may not always be clear in real systems. A basic distinction derives from their *order*.

4 (Discriminating power of order) *If we carefully consider the example 7.1, we shall see that the* goal *of level 1 acts as a* root objective*. This* goal *is actually an expression of a particular problem to be solved by the architecture.*

In fact, the manual [LBCC99] develops a similar example to the one in this text, for the particular problem of SOAR controlling a robot ordering cubes on a table. This problem, conveniently formulated in the architecture syntax, becomes goal *of level 1.*

We might realize that this goal*, although acting as a* root objective *during the resolution of the problem, could eventually be changed by the programmer. It is therefore, not a* root objective *in reality, but an* intermediate objective *of very high order.*

To find the actual root objectives *of SOAR, one would have to go for its ultimate purposes, which are related to general–scope problem solving. These are the objectives which justify that the architecture behaves as it does, and that it is built as it is.*

A more precise notion of the *time–scope* of an objective is the duration of the period of time during which the organization of the system is adapted to it. In other words, it is the duration of the period during which the objective *directs* the system behaviour. To refer to the evolution of the system in relation to the objective during this period, we shall call this period the *activity* of the objective. When we want to refer to the actual duration of the period, we shall use the term *time–scope*.[7]

5 (Time–scope) *Returning to example 7.1, we see that the time–scope of* goal *at level 1 is the sum of a collection of periods:*

- *time for selecting* operator *o1.2,*
- *time for detecting the* impasse*,*
- *time for generating the* goal *at level 2,*
- *time–scope of this goal,*
- *execution time of* operator *o1.2.*

Selection of o1.2. implies mediatory processes of search and prioritization. It can be observed that the time–scope of the goal *at level 2 also includes selection of* operator*,* impasse *detection, etc.*

The *order* of an objective is a metric of its time–scope relative to that of the generative objectives. The order of root objectives is '1'. The order of local objectives of shortest time–scope is '0'. The order of all other objectives is in $[0, 1)$. We might realize that the order of a specific, intermediate objective is affected by three factors:

[7]We must realize that the *time–scope* of an objective is the time it takes the system to achieve it, *counting all factors that may affect the process*, that is: time required by the actuators, associated information processing, controllers, process prioritization and scheduling during the period, etc.

- Level of abstraction: Objectives of low level of abstraction correspond to states of physical quantities. On the other side, abstract objectives may depend on abstract, simpler objectives, which in turn depend on physical objectives. This chain of dependences might grow very long in real systems. Therefore, the relation between abstract objectives and physical quantities may grow very complex and require a sequence of phases to be achieved.[8] High level of abstraction is a factor for longer time scope.

  Usually, an objective of high level of abstraction will have a large objective structure of lower level objectives. We might realize that the objective is dependent upon this lower structure, which also infliuences its time scope.

- Organization: As we have mentioned, a particular objective corresponds to a set of system properties. These properties, in the environment, display a behaviour which converges to the objective following a certain dynamics. In general, different properties would cause a behaviour which would converge following different dynamics. Therefore, time scope is partially determined by the part of the organization associated to the obective.

- Scenario of operation: The actual dynamics displayed by a system depend critically on the environment in which it operates. A same organization in different environments may cause convergence at different speeds or eventually not converge at all to a given objective. Therefore, the environment and constraints of operation during the activity of a specific objective determine its time scope.

  As we might realize, the scenario of operation during the activity of an objective is a critical factor. It is undetermined, due to the uncertainty of the environment. Therefore, the order of an objective is undetermined and can only be estimated. Greater knowledge about this factor leads to more accurate and reliable estimations.

A general formalization of the notion of order can be provided as follows. Let the order of generative objectives be '1', '0' that of local objectives. Let ts be the temporal scope of a specific objective, and its level of abstraction al. If we denote the order function as Θ, we have:

$$
\begin{aligned}
N &= \{\theta, \quad \theta \in \mathbb{R}, 0 \leq \theta \leq 1\} \\
TS &= \{ts, \quad ts \in \mathbb{R}^+\} \\
AL &= \{al, \quad al \in \mathbb{R}^+\}
\end{aligned}
$$

$$
\begin{aligned}
\Theta : TS \times AL &\rightarrow N \\
(ts, al) &\mapsto \theta
\end{aligned}
$$

[8] An example of dependences between physical levels and abstract levels can be found in the description of control architectures and the control pyramid in section 5.4.

We must remark first, that this formulation requires developing metrics for objective time scope and level of abstraction. In addition, the actual Θ function must also be defined. These three aspects are arbitrary to the criterion of the system designer or analyst. Self-adaptive systems may develop their own Θ functions by learning and self-modelling.

We must also remark that the order of an objective is a measure relative to a specific system at a particular instant of time. The order of an objective does not admit direct comparison with that of an objective of a different system or at a different instant. Let us consider that convergence to O is achieved in a time ts_O, and that we assign O an abstraction level equal to 1. Then the function of order for the objectives of this system will compare any other objective to these values, which have been chosen for this particular case. Comparison with objectives of other systems would require a common reference based on their respective convergence times and abstraction levels.

Finally, we insist that ts_O is usually unknown. Therefore the order function has to be based on an estimation of it. This forecast must be based on the current set of objectives, the structure and the program of the system at a particular instant. They define the estimated trajectory of states of the system, and define the performance of the system, the two aspects necessary for calculating convergence time. The accuracy of the resulting forecast will depend on the deviation between the estimated evolution and the real one, subject to perturbances and uncertainty.

7.3.5 Morphology of Objectives

As we have mentioned previously, an objective is a desired state of the pair (S, E) which can be complete or partial (see section 7.1.)

Objectives might not exist explicitly in the system. For example, it is usual that the root objective of artificial systems (generally the purpose of the system from the designer's point of view,) is embedded in their real structure. In this case, we shall say that the objective is *implicit* in the system. If there exists a representation, the objective will be called *explicit*.

6 (Implicit objectives) *Example 7.1 shows a case of* implicit *objectives. As it has been previously commented, the* goal *of level 1 is really an intermediate objective of very high order, while the real,* root objectives *are the actual purpose of the designer. These objectives are* implicit *in the system structure, and are what make the architecture adequate for solving problems (the actual* root objective*) and inadequate for other purposes. This fundamental adequacy of the system shows its basic* directiveness *towards an objective.*

An objective can be defined in two ways. First, by defining a *target* or *reference*, which is a representation of the desired final state (S, E). Second, by specifying a *set of constraints* for the values of system or environment quantities. An objective may also be defined by both a target and a set of constraints.

We shall say that an objective is a *target* or *setpoint* if it is defined as a target. In this case, the objective imposes no constraints on the organization, or on the dynamics of the system associated to the objective. Therefore, the actual organization and the process to realize it are undetermined. The undetermination must be resolved dynamically.

An objetive may consist of a set of constraints, and include no specification of a possible target. Constraints equal to time–invariant relations between the quantities of the system or the environment, which the system must comply with during the activity of the objective. The final state reached and the sequences of states which the pair (S, E) may actually follow during operation are not determined a priori, only bounded within the limits given by the constraints. In this case, we say the objective is *open*.

On the other hand, an objective may be specified both with a *target* and with a *set of constraints*. In this case, the ultimate state is given by the target, while the dynamics of the system in the process of achieving it is partially defined by the constraints. This will be called a *closed* objective.

7 (Basic control problem) *Example 7.2 describes a classical control system based on a PID controller. The desired behaviour of the system, to be achieved by adding the PID, is specified in terms of a* reference *and a* set of constraints*. It consists in keeping an output signal at the reference value. The constraints specify* how *the system is to behave while striving for the target.*

Now let us part from these considerations in order to distinguish a fundamental morphology of an objective. As we have mentioned, the activity of an objective is the period during which the system organization is directed toward it. In other words, the organization is configured corresponding to the objective, and causes a coherent behaviour. The objective is therefore mapped onto the system embedded in a real scenario of operation. In this case, the objective is *instantiated*, for the conceptual, desired state it stands for corresponds to real quantities of the system. Accordingly, we say the objective exists in *real* form. When we want to refer to the state in which an objective finds itself, we shall use *instantitated*. When we want to refer to the dynamic aspect of being instantiated, we shall say it is *activated*, ie.: its having an *activity*.

An objective, however, may eventually be *inactive*, in other words, not determining the behaviour of the system at present. In this case we shall say it is in *abstract* form. Objectives in abstract form are part of the system knowledge. They may be generated by problem solving, planning or other processes in the system, or they may be set by the designer in artificial systems.

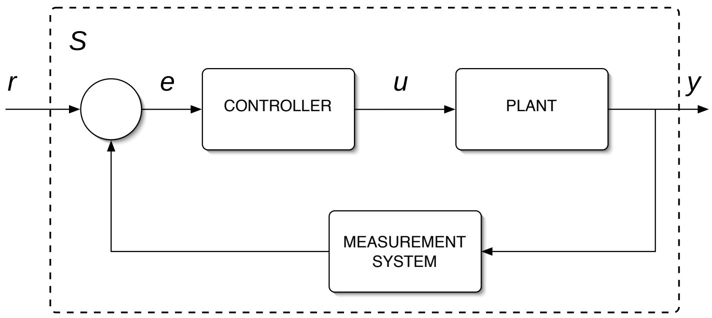
There are two ways in which objectives may exist in abstract form. An objective is *coded*, when it refers to actual quantities of the system, but is not currently active. A coded objective may be directly instantiated and become active. An objective may also be in *essence*. In this case, the objective does not necessarily refer to actual quantities of the system. It may be the result of abstract problem solving, or an inference process expressed in terms which may not correspond directly to the organization of the system. In order to become active, an objective in essence has to be coded first.

In summary, an objective in the system may exist in essence, coded or both, in which case it is in abstract form. It may also be active in the system, therefore instantiated and in real form:

$$\begin{array}{rl} \textit{abstract form} & \begin{cases} \text{essence} \\ \text{coded} \end{cases} \\ \textit{real form} & \{ \text{ instantiated} \end{array}$$

As we mentioned previously, the objectives can exist in two possible modes: *explicit* and *implicit*. The explicit mode occurs when the system has an abstract representation of the objective, regardless the objective is instantiated. The implicit mode occurs otherwise.

Example 7.2 (Targets, constraints, open and closed objectives)



Consider system S of the figure above. The system represents a typical feedback control loop, in which y stands for the system output, r for the input, e for the error signal, u for the control signal.

Signal r stands for the desired value of output y, e for the difference between r and y. u is the signal input to the plant. With adequate values of u, the plant will evolve until its output will eventually reach the required value r.

The typical controller in industry is a proportional-integral-derivative controller, abbreviated by PID. Its operation is described by the following expression:

$$u = K_1 e + K_2 \frac{de}{dt} + K_3 \int_0^t e \cdot dt$$

The designer adjusts the values of K_1, K_2 and K_3 in order to make the system comply with performance requirements. Typically they are:

- C1. Reaching the value y_r, corresponding to an input r, in a certain time since a sudden change in its value.
- C2. Not exceeding a maximum value, $y < y_{max}$.

- C3. That the final value, y_r is within a range from r:
  $y_r \in (r - e_{max}, r + e_{max})$.

Let us adopt a systemic perspective. We may observe that the objective of the designer is expressed by the constraints C1-C3 enumerated above. In view of these constraints, the designer defines the UC-structure for the system: chooses the PID-type controller, a particular measurement system, the comparator for calculating e, and assembles them through the adequate couplings, forming a feedback topology. The ST-structure is defined by setting K_1, K_2 and K_3.

The designer finds that the system must operate explicitly with the value of the reference signal, r. If a fixed value of r had been imposed, it would have been possible to make it implicit in the structure.

We might observe that the objective of the whole system, S, is closed, because it consists of a target (signal r) and constraints (C1-C3.)

These considerations refer to the whole system, S. If we consider the PID controller separately, as a system on its own, we can see that its objective is to achieve a null e signal. The objective is, however, implicit in its structure. We may observe that it realizes the constraints C1-C3 through its fixed parameters K_1, K_2 and K_3; no target is provided. Therefore, the objective of the PID controller is open. ■

7.3.6 Objective Dynamics

As we have mentioned previously, the objective structure of a system exhibits a certain dynamics as a result of the achievement of its intermediate and local objectives, and the generation of new ones (see example 7.1.) In this section, we are going to introduce the basic notions describing this dynamics.

The dynamic aspects of the life of an objective are given by four types of processes:

- *Generation*.
- *Activation*.
- *Activity*.
- *Deactivation* or *conclusion*.

Objective generation refers to the process by which either the essence or the code of an objective (or both) are generated. Activation of an objective stands for the process of instantiating an objective which exists in abstract form. Instantiation consists in adapting the organization of the system to the objective.

The activity of the objective is the evolution of the system during the time in which the objective is instantiated. It is understood that, during this period, the organization of the system corresponds to the objective, and so there exists directiveness toward it.

Eventually, an objective may be reached. We shall say that in this case, its activity *concludes*. However, a second objective might be instantiated before the conclusion of the first, overriding its organization. In this case, the first objective is *deactivated*.

8 (Conclusion of objectives.) *Let us consider example 7.2, and imagine that the user suddenly changes the value of the reference signal,* r_1*, to a different value,* r_2*. Subsequently, system will react in order to produce the corresponding output.*

When it is finally achieved, we may observe that the objective does not actually conclude*, because it is not actually* achieving *the new output, but* maintaining *it.*

In summary, the normal phases in the life of an objective are the following:

$$\text{generation} \rightarrow \text{activation} \rightarrow \text{activity} \rightarrow \left[\begin{array}{l} \text{deactivation} \\ \text{conclusion} \end{array} \right.$$

In artificial systems, an objective might be generated by the designer or by the system itself and stored in the system memory. Then, it might remain there for an undetermined period of time, until it is eventually instantiated, as a result of a planning algorithm. Consider, for example, goal 2 of example 7.1. When the impasse is reached while goal 1 is active, the architecture generates goal 2, whose objective is to resolve the impasse. If the system had no knowledge about the causes for the impasse, goal 2 would have to be generated dynamically, by analyzing the current situation. If the causes for the impasse were systematic and learnt by the system, goal 2 could be stored in memory. It could then be retrieved every time the situation would be repeated. We may observe that in this case goal 2 would exist in the system memory in abstract form. It would be instantiated at need.

Activation of an objective may consist in fast processes such as loading values in memory, or complex, time consuming processes involving physical aspects of the system, such as shifting a gear in an automatic gearbox.

As it has been mentioned, the activity of the objective might cease when the objective is achieved, or it may be deactivated by the instantiation of a new one, for example, by shifting to another gear.

The sequence generation–activation–activity–conclusion/deactivation is called the *general lifecycle of an objective*. It represents the phases of the life of an objective.

7.4 Organization

In this section we are going to analyze the autonomous system from the point of view of its organization, in relation with objectives and behaviour.

7.4.1 The System in Terms of Objectives, Organization and Behaviour

In order to analyze the organization of a system, it is necessary that we build an overall perspective of how it works in terms of the three main concepts defining its evolution: objectives, organization and behaviour.

As we have mentioned, at a certain instant of time the *organization* of the system corresponds to a certain *objective structure*. This means that the *properties* of the system make its parts evolve toward the objectives. We call *behaviour* to this evolution, which results of combining the properties of the system and the environment.

We have seen that there exist different kinds of *objectives*, *organization* and *behaviour*. *Root objectives* correspond to a set of properties of the system which constitute its *real structure*. When the system evolves in an environment, the *real structure* is the cause for the part of its behaviour which we call *permanent behaviour*. We understand that the *real structure* of the system stands for an intrinsic, basic fitness or adequacy of the system for its generative objectives. The *finality* of the system in the second sense mentioned in section 7.1.

Let us return to the GST concepts of *UC–structure*, *structure of universe of discourse and couplings*, and *ST–structure*, *the state–transition structure* (see chapter 6.) We may understand the *organization* of the system in terms of these two concepts. The first explains the *elements* forming the system and how they are are related to each other. The second, the possible configurations they could adopt.

In these terms, *permanent behaviour* derives from fixed arrangements of *elements* given by the *real UC–structure*, within a set of possible alternative configurations included in the the *ST–structure*. As it has been mentioned above, *permanent behaviour* follows a basic trend of the system to evolve towards its *root objectives* (see section 7.1,) the basic adequacy of the system.

However, changes in the *intermediate objectives* of the structure may eventually occur, causing some of the elements that form the system to be substituted for others eventually, or arranged in different ways. This reflects on the set of potential states reachable by the system. They constitute the variable parts of the *UC* and *ST–structures*, and derive in a specific *relatively permanent* or *temporary behaviour* when the system operates in the environment.

As we advanced in section 7.3, the dynamics of the objective structure and the corresponding organization represent the adaptivity of the system. This results from the changing of the properties of the system, allowing it to compensate for the perturbances derived from environmental uncertainty.

7.4.2 Functions

In order to explain the operation of a system, we shall use the concept of *function*. This concept will be defined in terms of the notion of *program*, which we shall develop following.

We shall assume that the dynamics of the *objective structure* result from the *mechanisms of directiveness* of the system. These mechanisms stand for resources and processes in the system which modify the *objective structure* and the *organization* correspondingly. As we saw in section 7.2, these mechanisms may be *structural* or *purposive*, but let us now consider them without distinction.

The *program* of the system is, by definition, the variable part of the *organization*. We may distinguish three kinds of programs in a system [Kli69, p.45]:

1. *Complete program* –instantaneous state together with the set of all other states in the system, and the set of all transitions from the instantaneous state to all other states of the system in time.[9]

2. *Subprogram*–instantaneous state together with a nonempty subset of the set of all other states of the system, and a nonempty subset of the set of all transitions from the instantaneous state to all the states under consideration in time.

3. *Instantaneous program*–instantaneous state together with the transitions from this state.

We shall understand a *function* as a succession of states associated to a particular objective. In other words, a set of states and their transitions, that contribute to realize a particular objective. A function is therefore, in the terms introduced above, a *subprogram*. Following the states and transitions of the *subprogram* will drive the system closer to the objective.

9 (Functions) *As we shall see following, a* function *might not always be* executed*, ie.: the* subprogram *it represents might not be followed. It can be only specified and kept in the system knowledge.*

As we shall see, the actual sequence of states followed by the system when attempting to execute a function might differ from the specification (we shall call this anomalous behaviour*.)*

In general, we may assume that a given objective may be realized in many ways, by different functions. Also, an objective may be realized by a single function or by a set of functions. The process by which an objective is assigned a function or a set of functions is called *decomposition*.

If we consider a function from a cognitive point of view, we shall say a function is an *algorithm*. An *algorithm* specifies a particular way of realizing the function objective. As we have mentioned above, the same objective could be realized in different ways, or, in other words, by different *algorithms*.

Algorithms may be stored in the memory of the system as representations, in which case they stand for functions in *conceptual form*. When the system is adapted and behaving according to a specific algorithm, we say it is a *grounded function*.

In order to build a first notion of function, we have parted from the concept of *program*. However, there is a point we must remark. As we have seen, objectives are equal to *global states* which we represented as pairs (S, E). This means that they specify a state or a class of states of both system and environment.

On the other side, we have already mentioned that a particular objective $(S, E)^O$, is defined relatively to the system. This means that the specified state of the environment E^O, will really be achieved when the corresponding state is reached by the system–environment coupling and related quantities (see *strict* and *subjective* states of the environment in section 7.3.) This makes that, in strict terms, $(S, E)^O$ stands for a certain state of the system quantities exclusively.

[9]We may observe that the ST–structure of the system is the set of all complete programs of all states in the system.

We may infer that the desired state of the environment, E^O, will be achieved by the action of the system modifying it until the desired state is reached by the coupling.

10 (Specification of objectives) *Let us return to example 7.2. We may realize that the objective of the system of maintaining the output y at the level corresponding to the reference signal, r, is represented as an error signal of zero, $e = 0$. In other words, that the measurement of the output, $m(y)$, is equal to the reference value: $m(y) = r$. We may observe that these quantities (r, y) characterize the coupling of the system with the environment.*

In fact, $m(y)$ is a dependent quantity (y is independent, but the measurement system —which determines the function $m()$— is part of the system) and r is a quantity of the coupling.

The following sections will analyze functions mainly from the perspective of a subprogram, although we might introduce them, intuitively, from the *cognitive* point of view, which will be developed further in part III.

We may distinguish three main aspects of functions which we may call *afferent*, *efferent* and *deliberative*. They stand, basically, for input, output and generic computing processes respectively. We could classify the quantities involved in the subprogram accordingly into afferent, efferent and deliberative.

The evolution of afferent quantities represent the input to the function. Intuitively, the perceptive component of the function. The evolution of the efferent quantities, the actual action carried out by the function. Action must be understood as changing the state of a certain number of quantities, not necessarily physical action. The evolution of the deliberative quantities stand for the calculations and intermediate operations for producing the output.

7.4.3 Algorithms and Grounded Functions

We have introduced *algorithms* as conceptual specifications of successions of states, and *grounded functions* as their actual implementation in the real quantities of the system. The correspondence between both cannot be assumed to be direct, for a grounded function is subject to restrictions derived from the actual scenario of operation and the resources of the system. Specifically, we may observe:

- A grounded function must refer to a specific set of quantities. The states contained in an algorithm may be undetermined. This means that the values of some of the system quantities may be unspecified by the algorithm.[10] Matching the algorithm with a specific set of system quantities implies resolving the undetermination for the set. In other words, the grounding process must assign the unspecified values. This may lead to emergent behaviour.

- An algorithm may or may not specify time. In the first case, the performance of the resources of the system in time must meet the specification, otherwise the algorithm will not be followed. The second case may be due, mainly, to two reasons.

[10] See *complete* and *partial* objectives in section 7.1.

First, that time is irrelevant for the algorithm, in which case grounding the algorithm is straightforward. Second, that time parameters are relevant, but unknown. This would require resolving the time specification for grounding the function.[11]

- The algorithms known to a system may come from many sources, for example: database programming, learning, or abstract problem solving. Due to this, it may occur that the algorithm does not perfectly match the case of application. This may have several effects. The most representative example may be that the specified states cannot be actually reached due to insufficient range in the real system quantities.

- The specific set of system resources for grounding an algorithm may be also necessary for other current algorithms. This would imply need for resource–sharing or prioritization. Inefficiency in managing shared resources will most commonly lead to emergent behaviour and failure of time–specifications.[12]

7.4.4 Functional Structure

As we have mentioned previously, functions are associated to objectives. The current objectives in the objective structure will therefore have a set of associated, grounded functions which we shall call *functional structure* of the system.[13] Ideally, functions and objectives will match perfectly. In reality, the resources upon which functions are grounded, and the circumstances in which they operate may make them differ.

Objectives are decomposed into one or more functions that realize them. Therefore, we can understand that this correspondence between functions and individual objectives defines a correspondence between the *objective structure* and the *functional structure* of the system.

As a single objective may decompose into multiple functions, the correspondence between structures will not be direct in general. This means that, although there will always exist a certain relation between dependences among objectives and dependences among functions, they are not necessarily equal. Also, we may realize that grounded functions are associated to specific resources of the system, which implies there may exist dependences between the quantities, imposed by the substrate, which might not exist in the objective structure.

As we have advanced, the adaptivity of the system is based on reflecting the objective structure into the functional structure. As we have advanced, we shall generically

[11] Notice that any undetermination in an algorithm, regarding time or any other aspect, may eventually lead to *emergent behaviour*, ie.: not foreseen by the designer or contemplated by the *mechanisms of directiveness*.

[12] The issue of resources in relation to functions is becoming increasingly important. [WLC00] assesses the issue of resource requirements. It proposes a methodology to evaluate performance and resource requirements depending on the system functions in the SOAR architecture [New90, RLN93, LBCC99] (by adding them incrementally and measuring resources and performance.)

[13] This association may be intentional, as in the case of artificial systems, or *accidental*, as in the case of biological systems. In this last case, the very nature of system objectives is matter of intense debate in biology, sociology, law, religion, etc.

call this process *functional decomposition* of the objective structure, or of a particular objective.

Functional decomposition represents a factor for system autonomy. We may assume that, in general, a given objective structure may be realized by different functional structures. In other words, different arrangements of system elements and successions of states may realize the same set of objectives.

This means that failing to achieve a particular intermediate objective may lead to two possible alternatives in order to maintain convergence:

- First, to redefine the objective structure in order to maintain finality.
- Second, to implement a different algortihm by executing a new functional decomposition, maintaining the objective unaltered.

Reconfiguration of the objective structure, functional decomposition and selection between both constitute the core mechanisms of directiveness of the system.

7.4.5 Anomalous Behaviour

As we have already mentioned, the *behaviour* of the system is the result of its grounded functions operating in an environment. Let us develop this basic idea by returning to the notion of function as a subprogram.

A function is a particular succession of states with a set of associated transitions. Evidently, these states must be among the possible states of the system in order for the function to be grounded (see the notion of *complete program* introduced in page 123.)

When a system is executing a grounded function, it follows the specified succession of states *automatically*, initiating the next transition once each state is reached. This dynamics is specified in the algorithm, as shown in figure 7.2 (a). Ideally, the grounded function is implemented in such a way that only the specified states and transitions may occur, following (a) of the figure.

This is not the general case in real systems, due to the factors mentioned in section 7.4.3, that appear when the algorithm is grounded.

The dashed lines and arrows in figure 7.2 represent states and transitions which are not specified in the algorithm. Let us consider case (b). When the conceptual function is grounded, the real quantities of the substrate might not admit the same range of values than the conceptual ones, or have different time performances. Considering S2, this may cause that S3 cannot be achieved directly, and that S7 is achieved instead. If the structure of the resources corresponds to the objective of the function, it should drive the system back to the specification, as shown by the transitions S7-S5-S3. This case is also shown by S1-S8-S2 and by S1-S5-S3.

However, this might not always occur, so that, eventually, reaching unspecified states could make the system diverge from the algorithm, as in case S3-S6-S10–. Abandoning the specification can also drive the system to undesirable stable states such as S9, in which the system would remain.

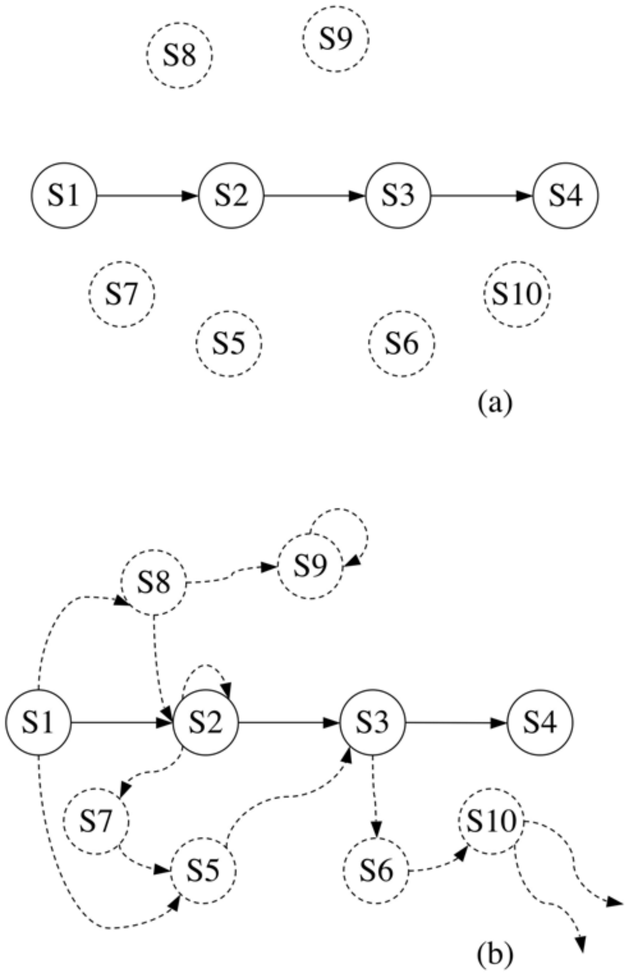


Figure 7.2: Normal and Anomalous Behaviour. S1-S10 are possible states of the system. (a) represents the specification contained in an algorithm, referring to a succession of states, S1-S4, and their transitions. Ideally, the grounded function would drive the system through this succession. (b) represents several examples of anomalous behaviour when the function is grounded, due to the constraints derived from the substrate.

Let us summarize the main causes for anomalous behaviour mentioned in section 7.4.3. We may observe that there are two main kinds: those due to undetermination of the function specification in the algorithm, and those due to substratal constraints. We have seen an example of the latter in the system reaching S7 instead of S3.

As an example of the consequences of undetermination, we may observe that, according to the algorithm shown in case (a), the only transition for S1 is S1-S2. However, we see in (b) that this is not so in reality, for S1 has two additional possible transitions. This difference is due to an incomplete specification of S1 in the algorithm. In other words, that the algorithm considers only some of the quantities really defining S1. The *unspecified* quantities are responsible for the other two possible transitions from S1. The cases of S2 and S3 are analogous.

Let us mention briefly how knowledge might eliminate or damp anomalous behaviour. Better knowledge means better algorithms, in the sense of higher accuracy in state and transition specifications on one side, and better modelling of the substrate, meaning algorithms better adapted to its real characteristics. On the other hand, better knowledge also means better processes of functional decomposition: higher accuracy for algorithm selection, improved correspondence between algorithm and substrate.

7.4.6 The Node–Stream Model

The node–stream model is a tool for modelling the functional structure of autonomous systems. It is intended to make the structure of functional dependences explicit. It will also make explicit the three cognitive aspects of functions, which we have introduced previously as afferent, efferent and deliberative. In this way, dependences between functions and dependences between cognitive subprocesses will be shown.[14]

We may distinguish two kinds of dependences within the functional structure. On one side, let us consider an objective which is decomposed into multiple functions, each defined by its algorithm. We may observe that these algorithms will necessarily be related through the common objective. They all will contribute to it. More, these algorithms can share information (ie: values of conceptual quantities.) This is an example of the first kind of dependence, which we shall call *cognitive dependence*. It refers to conceptual relations among functions or parts of functions in the structure.

On the other side, grounded functions differ from their algorithms in a series of constraints, given by the substrate on which they are implemented. These constraints are time–invariant relations between the real quantities of the substrate (derived from the UC and ST–structures.) As to the resulting function, they are additional dependences to the cognitive ones derived from the algorithms. We shall call them, generically, *substratal dependence*, or occasionally *structural dependence*, as they derive from the structure

[14]Some phenomena in cognitive systems can be observed at different scales, from microfunctions related to local objectives to the global system function related to root objectives. In many cases, the same structure of operation is followed at different levels of aggregation, in a kind of *fractal* disposition. The node–stream model is intended to provide a basic tool for expressing such operation and organization in systems.

of the substrate.[15]

The model shows both aspects through the concepts of *node* and *stream*. *Nodes* will model the functional structure, which, as we have seen, has a correspondence to the objectives of the system, and therefore a direct relation with directiveness. *Streams* will model the cognitive elements of which nodes are composed. Using both concept in this way will show, on one side, the topologies of structural and cognitive dependences are shown. On the other, possible points of incompatibility: functions competing for the same resources, mutually excluding functions, etc.

We might realize that the system managing a model of its own functional structure according to the principles shown here could eventually prove a basis for its processes of directiveness. An example of this will be shown in the commentary of a fault-tolerant system in chapter 14.

Nodes

A *node* represents a grounded function. In other words, a time–invariant relation between specific, real quantities of the system, or, in general, a set of time–invariant relations. We shall use the term node or *functional unit* indistinctively, although generally the first will denote a grounded function within a structure, and the second, in isolation.[16]

Therefore, we can understand that the functional structure is a topology of nodes, in which each node depends on others. Dependences may be of any kind. For example, hierarchical in the *client–server* or *master–slave* senses or *substratal*, as in resource–sharing functions.

A node in its general form is composed by four elements which we shall call *node elements* or *node components*. They are conceptual elements which refer to the cognitive nature of their operation,[17] and may not necessarily correspond to separated sets of resources. These elements are:

- Afferent.
- Efferent.
- Deliberative or core.[18]

[15]In advance to aspects developed further, we may comment that inconsistencies between cognitive and substratal dependences, restrictions of cognitive dependences imposed by substratal dependences, and, in short, any restriction to the function algorithm derived from the substrate is a negative factor for autonomy.

[16]The concept of node proposed here is a generalization of notions from different sources. In particular, we want to mention the RCS architecture, which proposes a parallel notion of node, also based in cognitive considerations [Alb99, Alb95, Alb91, GMP+01]. Essential concepts about concurrent processing and related topics have been studied in several sources, especially [Sta98, BW97, RM86, MR86].

[17]The terms *afferent* and *efferent* have been chosen in honour to their use in the classic book by Herbert A. Simon, *The Sciences of the Artificial*, [Sim90, p.140].

[18]From now on, the term 'core' will be preferred, in order to avoid connotations derived from traditional uses of 'deliberative' as antonym to 'reactive' or 'automatic.'

• Integrative.

The *afferent element* stands for the processing of the values of input quantities. We may observe that the degree of development of this element may vary widely depending on the functional content of the node. Operations can range between data acquisition and complex perception. We can imagine the afferent component as the part of the function which transforms real, *grounded* information into symbols or values of conceptual quantities.

The *efferent element* plays the role of *output* of the function. It delivers the result either to the rest of the system or to the environment. In this last case, we could conceive it as the inverse to the *afferent*, in the sense that it would map symbols (values of *conceptual quantities*) into values of *real quantities* (ie: action.)

Conceptual operations are executed by the *core element*. They are explicit operations with symbols. Examples are, deduction, problem space search, inference, etc.

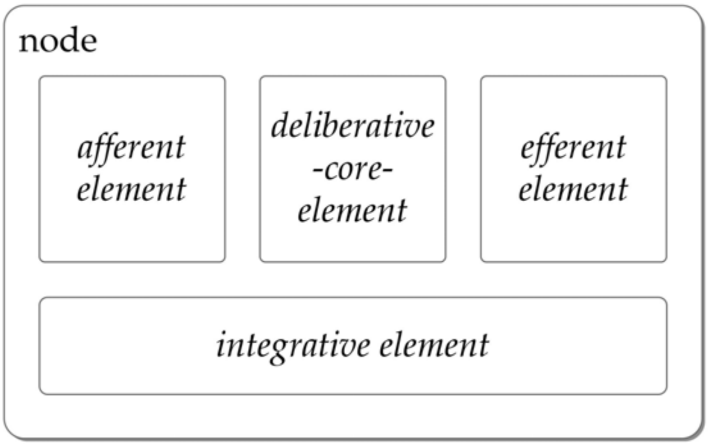


Figure 7.3: General Diagram of a Node.

Finally, the *integrative element* stands for the common, structural aspects of the *functional unit* (the *node*.) We may see the three afferent, efferent and deliberative components as subprograms in themselves, meaning that they actually stand for certain time and state specifications. From the point of view of states and transitions, the integrative element stands for a collection of subprograms which provide additional states, and couplings with the rest of the functional structure to the afferent, efferent and deliberative elements. From a cognitive or informational point of view, it provides mechanisms of synchronization, communications, resource–sharing and all other structural aspects to the other components of the node.

We may observe that this idea of the node elements represents the most general case, in which the node may have highly developed elements. However, it may occur that specific nodes in the system do not perform either afferent, efferent or deliberative tasks. In such cases, the corresponding element is said to be *degenerated*. This means that a node which exclusively performs perceptive tasks will have degenerated deliberative and efferent components. Analogously with the remaining possibilities.

It must be mentioned that a system composed by nodes exploits composition of node states to generate the global system state. System memory is implicit in the nodes. The four elements have access to a set of representation resources for their operation. These resources are used for intermediate quantities and for symbolic representation.

In order to analyze a system in terms of nodes, we must realize that a node is a conceptual structure.[19] This implies several points concerning realization. First, as it has been mentioned, afferent, efferent, deliberative and integrative elements may not appear cleanly separated.

Second, that the particular way of modelling a system with nodes is arbitrary. We could consider nodes at different levels of generality and abstraction in the system. For example, we could model the system by a single node, explaining its global behaviour. Oppositely, we could consider each individual time–invariant relation in the system as a different node. Intermediate levels of grouping could also be adopted. More, we could model the system overlapping the previous possibilities. In this case, we would arrive to a structure formed by nodes which would be formed, in turn, by simpler nodes. We shall call this *node nesting*. We understand this way as the most illuminating for its power of explaining simple systems as well as complex ones, and showing the relations between their parts.

As we have mentioned in previous sections, there exists a correspondence between the functional structure of a system and its objective structure. This correspondence may not be direct. However, an objective will be *decomposed* into a series of *nodes* (see section 7.4.2,) or in other words, realized by specific sets of resources following particular successions of states. We shall call the nodes associated to a particular objective its *functional scope*. The correspondence between objective and functional structures is therefore given by the topology of functional scopes.

We might observe that functional scopes may overlap, requiring coordination between the concurrent functions. Coordination implies two main aspects; on one side, managing resources to enable resource sharing. On the other side, resolving conflicts between overlapping nodes. Overlapping nodes may not always be compatible. The states specified in the algorithm of one of the overlapping nodes might not be compatible with the algorithms of the other nodes, on the same resources. In this case, a solution must be found according to the finality of the system. This may imply prioritization, new functional decomposition or eventually local reconfiguration of the objective structure.

11 (Resource allocation) *In chapter 14, a fault–tolerant system is analyzed. It is a multi–processor computer. When a certain component (processor, memory, communication link) fails, it may become useless,*

[19] A conceptual structure in the mind of the analyst or designer. Actually, the *node* model can be seen as an abstraction of other engineering models: the component–connector model used in software engineering, the unit–pipe model in chemical engineering among others.

forcing the system to search for free computational power in other components. If the system finds no spare components with which to execute the affected application, it will assign it to already–used resources, sharing their computational power.

In this case, the authors make no comment on a possible incompatibility for sharing resources between two applications. In other cases it could occur. Resource–allocation algorithms must take into account physical limitations, component capacity, integration implications, potential impact on performance, and other derived factors.

Streams

In essence, *streams* model functions, from a different point of view than *nodes*. While a *node* distinguishes four cognitive elements, a *stream* distinguishes operational/implementational elements. We shall see that these two points of view are related. A *stream* is formed by four elements, as represented in figure 7.4:

- Input interface.
- Output interface.
- Executor.
- Function definition.

Briefly, the functioning of a stream consists in the appearing of values in the *input interface*, which stands for the coupling with the environment, or with the rest of the system. The *executor* transforms the values of its quantities derived from the input interface, which represent the actual operation of the function. In logic terms, executor quantities stand for operators, buffers, registers, etc. Finally, the output is conveyed to the environment or to the rest of the system, through the output interface.

We may observe the analogy of *streams* with *nodes*. Cognitively, the *afferent element* of a node plays a similar role as the *input interface* of a *stream*; and the *efferent element* as the *output interface*. The *executor*, however, is not analogous to the *core* or the *integrative* elements. The executor refers to all the *grounded* operations between inputs and outputs. This includes part of the *afferent* and *efferent* elements (except their interfaces,) the *core* and the *integrative* elements. The *function definition*, on the other hand, is a *conceptual* entity detailing the resources and operations of the stream. It does not appear explicitly in the *node* model.

As it has been mentioned, the *function definition* represents the complete specification of the function that will be performed by the stream:

- Set of resources associated to the stream.
- Structure of the stream (interfaces, executor, possible states, transitions).
- Resolution level of operation.
- Relations between quantities (algorithms).

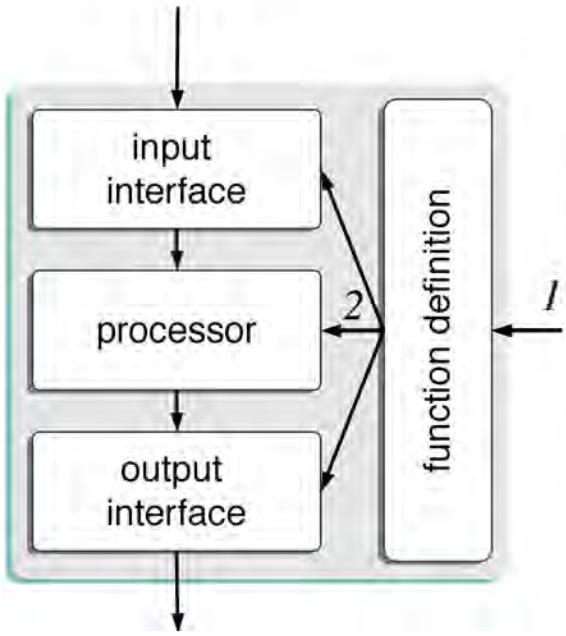


Figure 7.4: Diagram of the Stream Structure. We may observe the thin arrows indicating the flow of information during operation of a stream. The thick arrows indicate functional decomposition: 1–definition generation, 2–grounding.

The resolution level and the interface specifications are given by the functional structure, as they define the couplings of the stream with other streams and the environment. The set of resources and the algorithms are dependent on the availability of resources in the system and the objective associated to the function.[20]

Let us comment briefly on the process of functional decomposition. Functional decomposition involves three aspects: objectives, resources and algorithms. An algorithm in the system knowledge is selected in order to realize a particular objective with specific system resources. As we have seen, the result is a function definition, which constitutes a particular implementation of the selected algorithm, adapted to the resources of the system and to the objective structure in the current scenario. We may observe that the stream structure allows to model the function definition explicitly.

[20] We would like to remark that the stream notion is a generalization of the structures of the input and output channels in the BB1/AIS architecture, used as perception and action channels. The executor generalizes the *preprocessor* and the *channel driver* components of BB1. The function definition generalizes the *perceptual filter* and the *performance filter*, and the interfaces generalize sensors, actuators and I/O buffers (see that these last are the coupling of the channels with the blackboard.) For more information on BB1/AIS see [WB94, HR95].

We may observe that the system could be analyzed in terms of streams. We could summarize the whole operation of the system in one stream, or model each individual time–invariant relation, with intermediate levels of abstraction.

The Node–Stream Model

We are now going to propose a combination of nodes and streams to analyze functions in systems, although they could be used independently, as it has been mentioned. *Nodes* are appropriate for illustrating cognitive aspects of functions, because they explain them in terms of cognitive elements. *Streams* are representative from an implementational–taxonomical point of view, because they distinguish between *conceptual* and *real* components explicitly.

The node–stream model is a composition of the node and stream structures. Each of the node elements is modelled by a stream. Therefore, each node element has input and outputs interfaces, processor and function definition.

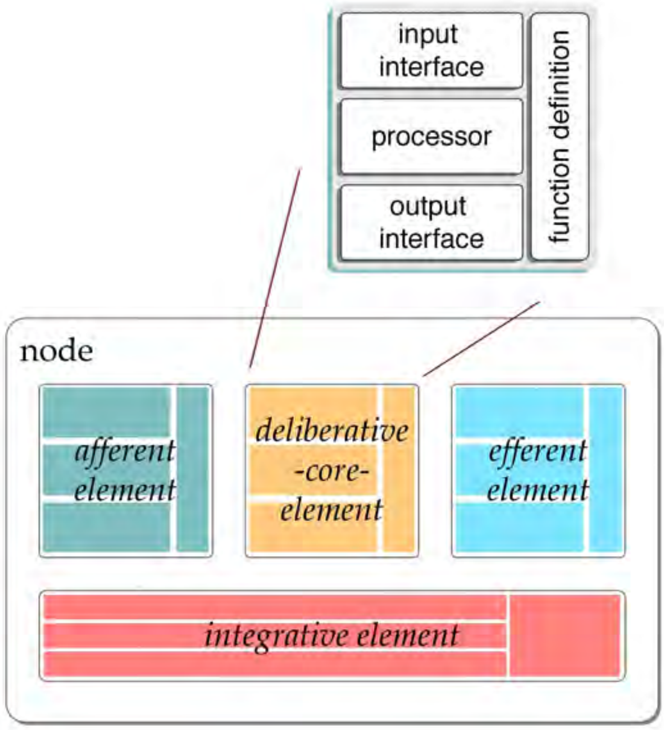


Figure 7.5: Node–Stream Model.

In order to illustrate the adavantage of composing nodes and streams, we must comment briefly on the relation between the node–stream model and system knowledge. As we have mentioned previously, functional decomposition produces grounded functions to realize an objective. This means allocating and configuring resources, so that their ST-structure adjusts to the particular algorithms considered as closely as possible.

Functional decomposition may involve selecting algorithms from the system knowledge or generating new ones dynamically. System knowledge may include algorithms which are specialized in afferent, efferent, integral or deliberative tasks. This knowledge may be combined arbitrarily when forming a node. In this way, the same afferent algorithm could be grounded with different deliberative, efferent and integrative elments, in order to form different nodes in each case. The same applies to any of the other types of algorithms.

We may assume, therefore, that algorithms could exist in knowledge independently from nodes. They would constitute function definitions for streams, so that a *node* would be composed of four *streams*. Nesting nodes and streams as proposed in the node–stream model, shown in figure 7.5 allows to make this explicit.

Let us analyze the interactions between the elements in the node–stream model, represented in figure 7.6. From a cognitive point of view, the node reads its environment with the afferent element, processes the information with the core element, and it executes action through the afferent element. The integrative element coordinates the three, so that the overall operation is coherent with the node objective. This sequence of processes stands for a sequence of changes in the node quantities represented by the thick arrow labelled '1' in the figure. The afferent and efferent elements act as the cognitive input and output interfaces of the node function.

It is common to conceive the afferent element as an actuator, executing physical action on the environment. This is a particular case of the notion proposed here. The afferent element outputs the result of the node function, though this might be physical, as in the case of an actuator, or conceptual, as in the case of a node specialized in deduction, inference or problem solving for example.

However, we must analyze the node from the point of view of its elements. The interactions between elements in this case are represented by thin arrows in the figure. Let us describe the two types of component–interactions indicated in the figure by arrows 2-3. Interactions of type 2 represent interactions between the components and between the node and the rest of the functional structure mediated by the integrative element. These interactions are therefore coordinated according to the integrative criteria of the node. We may observe that the cognitive interaction of the node, represented by arrow '1', may correspond to different combinations of interactions between components.

Arrows of type 3 represent interactions between the node elements and the rest of the functional structure which are not mediated by the integrative element. Therefore, these interactions are not coordinated with the operation of the node. They represent dependences between the elements and other parts of the system (including the other components of the node) which have not been modelled in the process of functional composition, and therefore not included in the integrative element. We shall see in section 7.5.3 that this type of interaction may be a negative factor for system autonomy

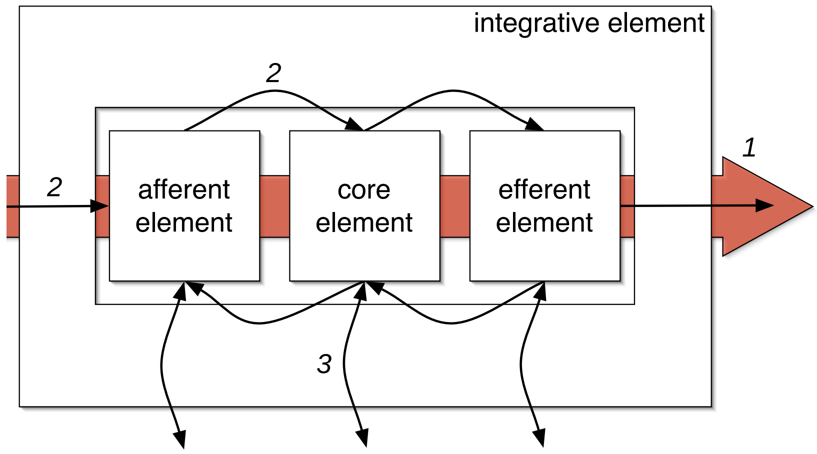


Figure 7.6: Interactions in the Node–Stream Model.

and that it should be minimized.[21]

12 (Encapsulation) *Interactions of type '3' affect the operation and dynamics of the node independently from the* integrative element*. This fact is a factor for* synergistic specificity *[SP05, p.282]: the elements achieve* synergy *with the rest of the system by being* specific *to a particular configuration of it. This may present some advantages regarding performance in that configuration. However, it contravenes two basic principles of* modularity*, and thus, is negative for* adaptivity*:* separability *and* recombinability *[SP05, p.283]. We shall assess these aspects in detail in the following sections.*

7.5 Autonomy

We may return to the practical senses of autonomy introduced previously, and summarize them in two:

1. Minimum dependence of the system from its environment.
2. System cohesion.

[21] The *node–stream* model could prove useful in system synthesis as a commonality in structure and in partial isolation. It helps in the process of system structuring (objective synthesis and function grounding.) Analyzing systems according to the *node–stream* model, however, may become a daunting task, maily because of the fact that their *accidental* architecture would derive in large quantities of overlapping grounded functions.

We may assume, without loss of generality, that achieving system objectives is implicit in these two senses.

In the light of the previous sections, we may conclude that the effective autonomy of a system relies on the robustness of is purposive and structural directiveness. This means that the mechanisms of structural and purposive directiveness generate adequate organizations in the system. As we have seen, system organization results in a state–transition structure. The functional structure of the system, result of its mechanisms of directiveness, consists in subprograms which drive the system to its objectives. We have seen that different factors may lead to anomalous behaviour which are out of the specifications of the functional structure. This mismatch between functions (organization) and actual behaviour of the system stand for loss of efficiency in convergence, and eventually for divergence or system instability (understood as loss of system cohesion, see section 7.5.2.)

Intuitively, the degree of autonomy of a system stands for the scope of *intensive* and *qualitative uncertainty* (see section 5.2, p.76) in which the system is capable of preserving convergence and cohesion.

In the previous sections, we have analyzed different aspects of systems related to autonomy separately. We have discussed *finality*, *objectives*, *directiveness* and *organization*, and identified several points of relation with knowledge. In this section we shall try to build a unified vision, assuming knowledge and reconfigurability as a basis for system autonomy.

7.5.1 The Cognitive–Grounded System Model

The cognitive–grounded system model is an ontology which serves as a background for explaining global aspects of the operations of autonomous cognitive systems. It will be abbreviated by CGSM.[22]

The basic idea of the model is to conceive an autonomous system as a duality of a *cognitive system*, CS, and a *grounded system*, GS. As we have seen in the introduction to GST in chapter 6, real quantities refer to those which actually exist, while conceptual quantities refer to those which are assumed. We may observe that both types of quantities are involved in the operation of an autonomous system based on knowledge. It is useful to separate both types of quantities. The conceptual operations and knowledge (quantities) form CS. The physical quantities and their dynamics constitute GS, see figure 7.7.

We may assume that there exists certain independence of operation between CS and GS, so that operations in CS may not necessarily cause a change in the state of GS. This, from the point of view of autonomy, provides the system with degrees of freedom of action, which stand for its capacity for reacting to environmental uncertainty. As we have seen throughout the text, knowledge is a factor in several aspects of directiveness and functions.

[22]The literature assessing related topics is vast. [Har90] is perhaps a first, modern account of the problem.

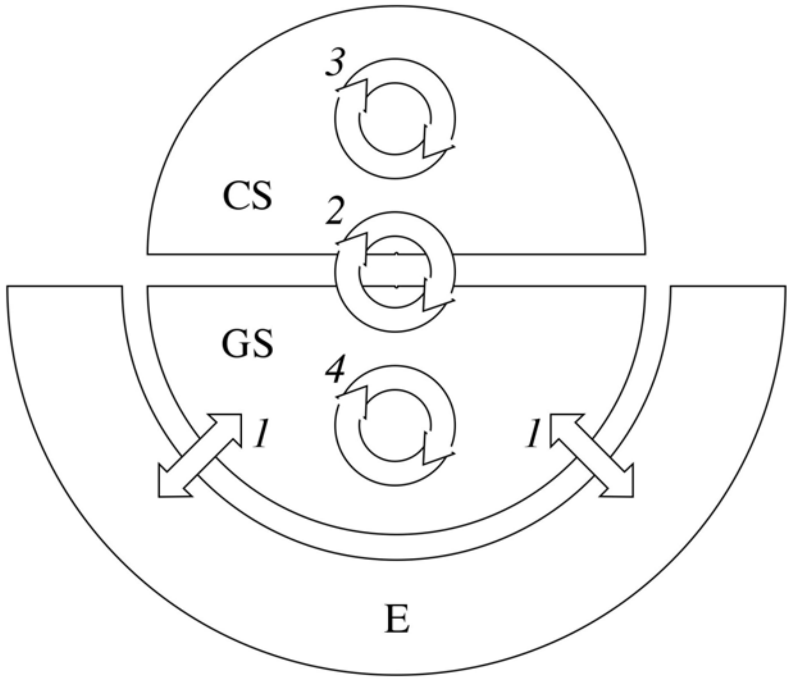


Figure 7.7: The Cognitive–Grounded System Model. CS–Cognitive System, GS–Grounded System, SE–System Environment. 1–interaction with the environment, 2–perception and grounding, 3–cognitive operation, 4–grounded operation.

Operation of CS and GS

Let us consider system operation following structural directiveness, in which there exists a defined functional structure. In these circumstances, the parts of the afferent, efferent, deliberative and integrative elements of the system functions involving conceptual quantities form part of CS.

The processes of purposive directiveness are intrinsically part of CS operation, as they involve symbolic representations of system, environment, objectives, and conceptual processes. The result is a conceptual representation, which is then grounded into GS.

As we have advanced, CS may operate separately from GS, although the separation might not be perfect. Some processes and knowledge tokens may take place in CS regardless the environment and the current state of GS. One example of this kind of operation in humans is abstract detached thought, or learning from experience.

Let us build a global notion of the operation of GS and CS. We may observe that the notions of finality, directiveness, objective structure and functional structure, commented in previous sections, explain the dynamics of the whole system (including GS and CS.) This stands for GS, and CS considered from the point of view of the physical quantities that serve for its *substrate*. However, CS must be analyzed separately in its conceptual dimension, as it constitutes a critical factor for system autonomy.

We may assume that CS consists in a set of quantities, which we shall generically call *cognitive quantities*. Some of the cognitive quantities may refer to the current system and its environment. We shall call them *instantiated quantities*. Other quantities may become instantiated quantities in the future, because they refer to possible scenarios of operation of the system. We shall call them *potentially instantiated quantities*. There may exist abstract quantities not referred either to system or environment, of value for cognitive operations, which we shall call *intrinsically cognitive quantities*.

As any system, CS admits analysis of its organization in terms of time–scope, as it has been done in the previous sections, derived from the analysis of GST (chapter 6.) However, it could be more illuminating to consider the properties of CS in terms of the conceptual value of its quantities. We may understand CS as a superorganization of more elementary organizations. We shall call *instantiated organization* to the properties associated to instantiated quantities. It represents and corresponds to the actual system, and therefore we may understand that this organization displays a conceptual image of the actual GS and its behaviour. It is a self–model. We shall therefore call *cognitive model of the system* to designate instantiated quantities and their organization.

We shall call *general knowledge* to potentially instantiated quantities and their organization. The organization of general knowledge represents the actual knowledge of the system. The organization of intrinsically cognitive quantities defines cognitive processes, buffer quantities, configurable registers, etc.

Cognitive–Grounded Coupling

We may understand that the dynamics of CS and GS are determined by a combination of explicit and implicit factors. The dynamics of the organization of the system as a whole, common to GS and the substrate of CS, is partially affected by the environment implicitly. On the other hand, the explicit output of perception are symbolic representations, which determine the explicit operation of the system.

We have already discussed functions in previous sections, which answer to the dynamics of GS. We may call the dynamics of CS from the conceptual point of view *cognitive operation*. We may distinguish between *instantiated operation*, which corresponds to the dynamics of the cognitive model of the system, and *non–instantiated operation*. This stands for the rest of CS, formed by potentially instantiated quantities and intrinsically cognitive quantities.

Cognitive operation is input by *perception*,[23] and its output is *grounding*. Perception produces informational content. Grounding, which we have already mentioned, stands for the realization of conceptual quantities. As we have mentioned above, the dynamics of CS is partially determined by implicit factors, which affect cognitive operation.

[23] Perception will be analyzed in detail in part III.

7.5.2 Autonomous Operation

Let us return to the two senses of system autonomy mentioned previously, adding *finality* explicitly in order to clarify the exposition:

1. Independence from the environment.

2. System cohesion.

3. Finality (directiveness towards objectives.)

Let us analyze the process by which the system loses cohesion and finality. We may assume that the uncertainty of the environment appears as *perturbances* to the system. *Perturbances* are represented as block–arrows of type 1 in figure 7.8. The program of

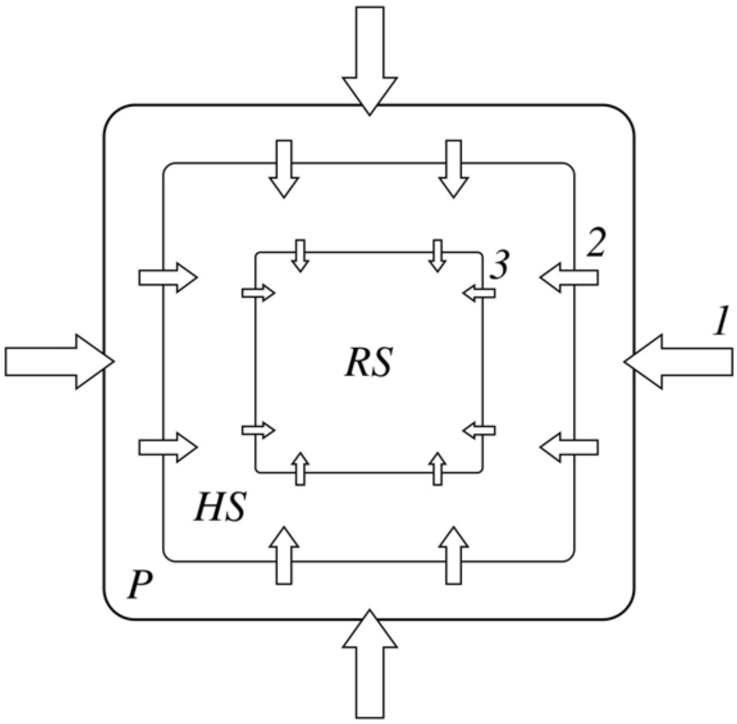


Figure 7.8: Propagation of Perturbances in an Autonomous System. The square represents an autonomous system. Intermediate squares represent the parts of the organization. RS–real structure, HS–hypothetical structure, P–program. 1–Perturbances to the system, 2–program failure, 3–structural failure.

the system, P in the figure, has certain capacity for compensating perturbances which we shall call *performance*. This stands for the actual efficacy of the local behaviour of the system. As we have mentioned, performance may eventually prove insufficient for compensating certain perturbances leading to what we shall call *program failure*.[24]

The consequences of program failure may affect the hypothetical structure, HS in the figure. At this level, directiveness mechanisms can operate in order to reconfigure HS for correcting operation. This implies modifying algorithms or entire parts of the functional structure. We shall call this *adaptivity*. We may realize from the previous sections on directiveness and objectives, that adaptivity may be *structural* in the case that it follows a function of the current structure, or *purposive*, in the case it is designed dynamically (we may assume that this implies symbolic operation.) In the event that adaptivity was not sufficient to recover convergence, this would lead to *structural failure*.

13 (Program and structural failure) *There exist many resons why the* performance *or* adaptivity mechanisms *of a system may prove insufficient for compensating* perturbances*. Artificial systems are designed to operate in certain scenarios. Neglecting the effect of age, the reason for* performance *or* structural failure *is that the actual conditions in which the system has to operate do not correspond to the original scenario of operation for which the system was designed. This may happen due to an inappropriate design or to the uncertainty of the environment.*

Intensive uncertainty *(p. 76) may produce a perturbance in the parameters of the system of such magnitude that it cannot be compensated for.* Qualitative uncertainty *may change the scenario of operation drastically, in a way that the* system program *is powerless.*

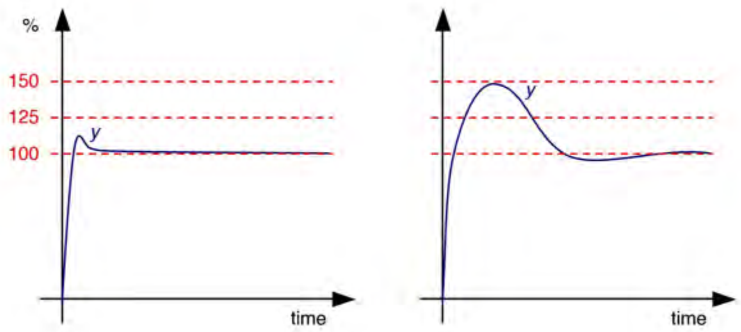
Example 7.3 (Performance and Adaptivity.)
Consider the classical feedback control loop of example 7.2, p.119. [GL00, p.5] offers the example of a *Current–controlled DC motor with Saturated Current*, following this loop, in order to show the effect of a saturation in the plant from the point of view of system *performance*.

The system output, y, is the angular position of the motor. In this example, the plant is formed by an amplifier coupled to the DC-motor. The amplifier saturates when $|u| \geq 1$. In saturation, the input to the motor remains constant at the value corresponding to $|u| = 1$. The controller is a PID controller such as that of example 7.2, adjusted for non–saturation.

Let us assume that the system, when abruptly changing its reference, should not exceed an overshoot of 25%; this is constraint C2 of example 7.2.

The system has two regions of operation: when the plant is saturated and when it behaves linearly. Operating with the described controller, the *behaviour* of the system is shown —approximately— for both regions of operation:

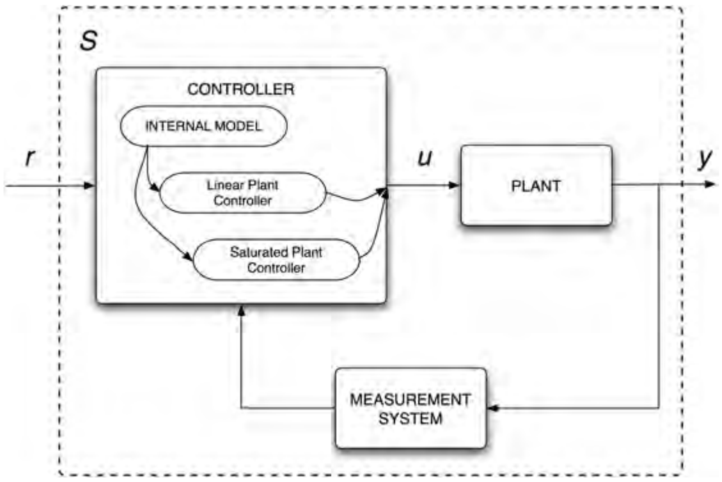
[24]The term *failure* has been insipred in fault-tolerant systems terminology, [Jal94], [Cri93].



We may observe that the overshoot does not reach 25% in the case of the plant operating linearly. As expected, the system would comply with the imposed constraint. On the contrary, the overshoot when the plant is saturated reaches 50%. In this case, there would exist *program failure*.

This system is not designed to provide any degree of adaptivity. All the structure of the system is therefore its *real structure*. There exist more developed control topologies than the classical feedback control loop considered which could be used for this case. [GL00] mentions internal model control, model–predictive control and parametric optimization among others.

Internal model control, IMC, is considered for an analogous system in [GL00, p.372]. The controller is substituted by a topology of components including a system model. The model is used to change the controller depending on the region in which the system is operating. Conceptually, the new topology can be represented as indicated in the following figure.



The internal model is used to enable the controller which is appropriate to the current situation, deduced from system parameters included by the designer; in this case possible parameters are signal u and the output, y.

This new system is capable of satisfying the constraints in both regions of operation, by applying an optimized controller for each case. This has been achieved by increasing —adding— a *hypothetic structure* to the system, and mechanisms of *adaptivity*.

The details of this system can be consulted in [GL00]. ■

Structural failure would propagate to the real structure of the system, as represented by block–arrows of type 3 in the figure. Structural failure stands for loss of system cohesion. The real structure of the system, as we have mentioned previously, refers to the main, constitutive properties of the system. They represent intrinsic aspects of the system regarding its elements, its topology, and its finality. Structural failure leads to alterations of these properties, and therefore affect the system itself. The effect of structural failure on RS will be called *degradation*.

Performance

We may see that system *autonomy* is equal, in a sense, to *performance* and *adaptivity*. *Performance* is a notion of the capacity to maintain convergent local behaviour against perturbances. Returning to the notion of function as subprogram, developed in section 7.4.2, we may realize that the performance of a function depends mainly on three aspects:

- Accuracy of the state specifications.
- Feasibility of the transition specifications.
- Completeness of the specification.

As we mentioned, the specification of a function may be based on undetermined state specifications, or define excessively demanding transitions. This could lead to anomalous behaviour, and eventually, to program failure.

In figure 7.2 in page 127, we may observe that states S5-S10 were not included in the specification, therefore leaving their corresponding transitions undetermined by the function. In the case that the system would leave the specified succession, the return would be at random. A more complete specification would specify transitions so that the system would tend to the main succession of states in case of anomalous behaviour, as the transitions S2-S7-S5-S3.

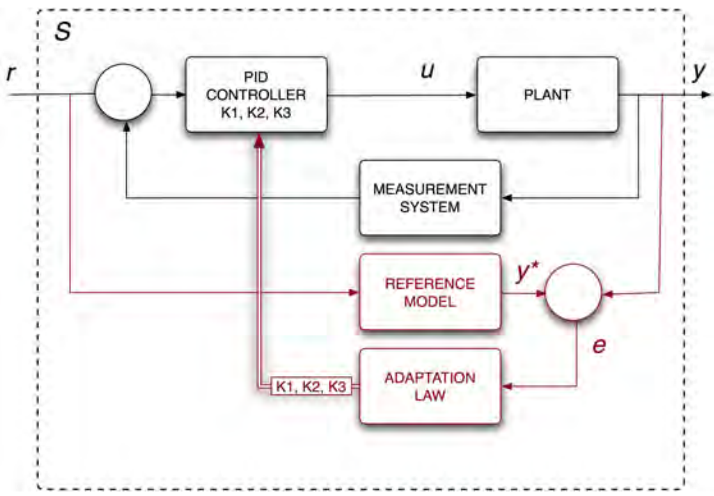
Example 7.4 (Changing performance)
Example 7.3 describes a system which adapts its structure to the region of operation of the system. In the example, the general technique of internal model control was mentioned. Model-reference adaptive control, MRAC [SL91, p.315], is a special case of this type of control.

Consider the classical feedback control loop of example 7.2. The K_1, K_2 and K_3 parameters define the PID controller. The designer calculates adequate values of these parameters for a given set of restrictions, C1-C3 of example 7.2, and a given plant, so that the system behaves as desired.

It is not uncommon that plants vary their parameters in different circumstances: when aged, when operating on certain ranges of their values or when the environment changes, for example. These parameters may be slowly–varying (in realtion with the variation of the inputs and outputs) or uncertain, or both. [SL91, p.311] gives some examples of such cases: "robot manipulators may carry large objects with unknown inertial parameters. Power systems may be subjected to large variations in loading conditions. Fire–fighting aircraft may experience considerable mass changes as they load and unload large quantities of water."

In such cases, a controller such as the PID of example 7.2, designed for a particular configuration of the plant, might not be valid when the parameters of the plant change. In fact, example 7.3 describes a case in which this change in parameters is abrupt (non–linear), due to the saturation of the amplifier to the motor.

Let us consider that the parameters of the plant of example 7.2 are slowly–varying. MRAC control could be implemented over the original control topology of the classical control loop as shown in the figure.



The addition to the original feedback control loop is represented in magenta in the figure. The reference model is used to specify the ideal system response, $y*$ in the figure, to the given reference, r. It plays a similar role to the internal model of example 7.3. The difference between the ideal response and the real response, e, is attributed by the system to changes in the parameters of the plant. An *adaptation law* specifies how the K_1, K_2 and K_3 parameters of the original PID controller must change in order to compensate for the changes in the parameters of the plant. For a given difference between real and ideal responses, e, it adjusts the PID with a new set of K_1, K_2 and K_3 parameters.

This system *adapts* by changing the parameters of the controller, which is always the same. The K_1, K_2 and K_3 parameters, part of the *real structure* in the original topology of example 7.2, have been made part of the *hypothetic structure*. The *program* of the system has been also increased. We may observe that the knowledge of the system has been increased with the incusion of the reference model and the adaptation law.

The operation of this system, however, is limited to the linear region, as in the first system of example 7.3. Saturation —a *qualitative* change in the conditions of operation— as in the second system, would lead to program and structural failure. A more complex topology, as the IMC, would be necessary to increase adaptivity in that case. ■

Adaptivity

We may see that *adaptivity* involves essentially two kinds processes: *objective configuration* and *functional decomposition*. As we have mentioned previously, functional decomposition is not uniquely defined, and different decompositions could be carried out for the same objective structure. We can consider that functional decomposition consists of the following phases:

- [Algorithm generation.]
- Algorithm selection.
- Grounding.

Algorithm generation is represented in brackets to express that it may take place independently from the other two. We could consider that adaptivity by *functional decomposition* may take place in four levels:

1. By maintaining a same level and re-grounding it, in order to improve its implementation for the actual scenario of operation.
2. By selecting a different algorithm from system knowledge, and then grounding it.
3. By generating a new algorithm dynamically and grounding it.
4. Finally, any of these alternatives, due to unfeasibility, may lead to re-definition of part of the functional structure. Major changes may imply changes in the objective structure. We shall call this *backpropagation of adaptivity*.

14 (Functional decomposition) *Examples 7.3 and 7.4 illustrate two ways in which system adaptivity is modified by* functional decomposition *of level 2. In both cases, the* algorithm *is changed. The first example shows a* qualitative *change, and the second an* adjustment. *The usual way in which the two systems are implemented makes the* re–grounding *level straightforward.*

However, there exist systems in which re–gounding *is of vital importance, while functional decomposition of level 2 is not implemented. An example of such systems are fault-tolerance systems. In these systems, the* algorithm *of the system must remain unaltered while the resources on which it is grounded may eventually change, due to the existence of faults. A particular fault-tolerant system is commented in chapter 14.*

Adaptivity by *objective reconfiguration* may be explained similarly to *functional decomposition*, in terms of three phases:

- [Objective generation.]
- Objective selection.
- Functional decomposition.

As in the case of algorithm generation, objective generation can take place independently from the rest, as well as triggered by events in the other two.

We may conclude from this insight into autonomous behaviour, performance and adaptivity that system autonomy is defined by: performance, standing for efficient algorithms, capacity for grounding, capacity for algorithm generation and capacity for objective generation. We may observe that they refer to two aspects of the system: the substrate on which it is implemented, and the knowledge and abstract processes it carries out, which we have conceptualized as CS.

7.5.3 Principles of Autonomy

Returing to autonomy in the system as a whole, we may conclude that there exist a short collection of factors for autonomy which enable high degrees of adaptivity. We shall call them *principles for autonomy*, in order to emphasize that they constitute principles of design of artificial systems. We may distil them as follows:

Minimal Structure: The organization of the system may be divided in two parts, program and structure. According to the *principle of minimal structure*, the structure of the system must be minimized for higher autonomy, which stands for maximizing its program. This equals, firstly, to maximize system performance. Secondly, within the structure, its stands for minimizing the real and maximizing the hypothetical structure. This equals to providing maximum adaptivity.

Ideally, maximizing performance equals to minimizing program failure.[25] This consists in increasing the accuracy of the state and time specifications of functions, and their completeness. Increasing performance means on one side that algorithm specifications are adapted to the system resources. On the other side, that system resources provide the required characteristics.

Maximizing performance equals to minimizing the cases of anomalous local behaviour. This means that the program of the system is capable of compensating for most cases of intensive and qualitative uncertainty.

Within the structure of the system, minimizing the real structure is equal to preserving system cohesion. Maximizing the hypothetic structure equals to increasing reconfigurability, a factor for system adaptivity.

[25]This is not a straightforward problem in real systems, especially in the artificial case. Greater program equals to higher probability of errors, as well as need for greater magement resources.

15 (Increasing the program and the hypothetic structure.) *We have seen in examples 7.3 and 7.4 two ways in which both the* performance *and the* adaptivity *of the system are increased.*

In the first case, the structure of the system was significantly changed with respect to the original system of example 7.2. It shows how two independent structures of control are integrated by addition of a knowledge component.

In the second case, the hypothetic structure is also increased, although the increase in the program is perhaps more relevent. The original structure of example 7.2 is maintained, but a strong knowledge component is added, which permits adapting it dynamically.

Encapsulation: This principle stands for two main aspects. First, minimization of the couplings between elements. Second, for the construction of interfaces, in order to *encapsulate* heterogeneous elements.

We may realize that encapsulation contributes to autonomy in several ways. First, minimization of couplings is a factor for minimization of structure. Second, encapsulation favours reconfigurability. Third, encapsulation favours the accuracy of algorithms and knowledge.

16 (Encapsulation in systems) *Encapsulation is a well–known topic in many branches of engineering. The term was extracted and generalized here from the field of* software engineering *because it has been explicitly and specifically treated:*

> *The technique of isolating a system function within a module and providing a precise specification for the interface to the module is called encapsulation. [BW97, p.73]*

One of the properties derived from encapsulation is modularity*, defined as:*

> *Systems are said to have a high degree of modularity when their capabilities can be disaggregated and recombined into new configurations, possibly substituting new capabilities into the configuration, with little loss of functionality. [SP05, p.282]*

The repercussion of encapsulation in system adaptivity is therefore clear in this sense. However, we have pointed out the importance of encapsulation regarding cognition. For the designer, encapsulated systems are more easily modelled than synergistically specific *systems [SP05, p.282]. Their well–defined interfaces and isolated functionality allows traceability of interactions between modules.*

This also —especially— applies to self–modelling, self–monitoring, learning artificial systems. Additionally, the models of encapsulated systems are more easily integrated in mechanisms of directiveness such as functional decomposition *or* objective reconfiguration*. They permit tracing interactions between parts through their couplings, and model–separability. A clear example of this quality is offered in the* FTMPS *system commented in chapter 14 [BAB$^+$95], [DVC$^+$94], [VDL$^+$94], in which modularity of its implementation resources is the basis for its algorithms of reconfiguration.*

Homogeneity: The principle of *homogeneity* is best understood if explained referring to the elements and couplings of the system. Homogeneity stands for similarity between system elements and couplings.

The UC–structure stands basically for the real and hypothetical structures, and therefore we understand that part of it is constant, and that the rest is reconfigurable. We have seen that the principle of minimal structure requires the constant part to be minimum. As to the remaining elements and couplings, similarity is a factor for interchangeability.

Similarity may not be possible. In this case, homogeneity may be increased by intermediate elements which enable indirect coupling of heterogeneous elements. These intermediate elements are generically called *interfaces*.

From the point of view of the CGSM, homogeneity may be considered in two other senses. First, as knowledge constituting a common resource for all the system. This means accessible to all elements of the system. Second, as to the elements of CS, for them having a common structure, as in the case of the elements of the UC–structure. In fact, reconfigurability in system elements stands for connectivity between elements of knowledge.

We may realize that homogeneity represents increasing system efficiency, in the sense of optimizing the use of its resources. Homogeneity of its elements, implying interchangeability, maximizes the available resources for grounding functions, as well as the possible ways of reconfigurating. Similarly, homogeneity of knowledge maximizes its potential scope of use and power of representation.

17 (Homogeneity of substrate) *The hardware layer of the above–mentioned* FTMPS *system, commented in chapter 14, is formed by arrays of processing nodes. All nodes consist of the same components. On the other hand, the software layer is structured in topologies of* processors *which can be either for control or for executing user applications. These components are logical, meaning that their mapping to actual hardware processors may not correspond.*

We may observe that homogeneity has been used in this system to achieve a very high degree of recombinability *of components [SP05, p.283]. The system is built only on three categories of components; the substrate only on one. Adaptivity is based not only on modularity —there could exist multiple types of incompatible modules,— but in the possibility of exchanging a component for any other in the system. In this way, the number of reconfigurations admissible for the system is maximized (a form of adaptivity.)*

Isotropy of knowledge: stands for the quality of presenting coherent meanings under different contexts of interpretation. We may realize that system knowledge is generated within a particular scenario. This equals specific *functional* and *objective structures* which partially define the knowledge acquired. We shall call *biasing* to this partial definition. We may understand that different conditions of operation produce different biasing. Perfect isotropy means that the content of knowledge is independent of biasing; lower degrees of isotropy stand for reusability of knowledge in different contexts from the one in which it was created.

18 (Isotropy of knowledge) *[KS05] provides a model of the cognitive modules involved in music perception, and a brief overview of perception and its neurophisiological couterparts in humans.*

Music involves processes —and therefore knowledge— which can otherwise be observed isolatedly, "making music in a group is a tremendously demanding task for the human brain that engages virtually all cognitive processes that we know about, including perception, action, cognition, social cognition, emotion, learning and memory." [KS05, p.578]

Knowledge in humans is massively re-used for different purposes and by different processes. For example, there exists a strong relation between musical and language knowledge

> *even individuals without formal musical training show sophisticated abilities to acquire knowledge of musical syntax [. . .] Interestingly, it appears that human musical abilities are important for the acquisition and the processing of language: infants acquire much information about word and phrase boundaries (possibly even about word meaning) through different types of prosodic cues (i.e. the musical cues of language, such as speech melody, metre, rythm and timbre) [KS05, p.582]*

Isotropy of knowledge can also be observed in the use of memory in music:

> *Structure binding requires working memory as well as a long-term store for syntactic regularities, and processing of meaning information is presumably tied to a mental lexicon (containing lexical-semantic knowledge), as well as to a musical lexicon containing knowledge about timbres, melodic contours, phrases and musical pieces.[KS05, p.582]*

We may realize that the same knowledge and the same representations may be used, by different processes, independently from their biasing. *It must be remarked that achieving isotropy in artificial systems is a non-trivial issue.*

Scale and scalability. The previous principles assess constitutional aspects of systems. *Scalability* refers to the capacity of the system to *grow*. At several points in the text the issues of knowledge, performance and adaptivity have emerged. Availability of resources is a key factor for all. Larger resources constitute a possibility for higher autonomy through more possibilities of reconfiguration and a larger program. They may also constitute a factor against autonomy by increasing the number of independent quantities and enhancing the real structure.

The *principle of scale* stands for maximizing the degrees of *minimal structure*, *encapsulation*, *homogeneity* and *isotropy of knowledge* by system *growth*. The term *growth* is used in order to remark that the process is a composition of increasing resources and directed resource integration.

7.5.4 Ideally Autonomous Systems

The notion of an *ideally autonomous system*, *IAS*, will stand for a conceptualization of a system in which the principles of autonomy are optimally realized.

We might realize that a system which is *absolutely* autonomous is impossible, if we understand it as capable of achieving its objectives in any circumstances, under total uncertainty. We may realize that this could mean impossible reaction speeds, instantaneous solving of uncertainty,[26] decision taking, reconfiguration and action. These requirements could only be achieved by a system having infinite resources and infinite

[26] Bounding, modelling, elimination.

knowledge. In this case any perturbance could be anticipated, characterized and compensated for.

Let us analyze the case of a non–infinite ideal system in which performance of resources and knowledge are maximal, which we shall call *ideally autonomous system*.

We must remark that the *principles of autonomy* stand for system optimization for autonomy. Nevertheless, there exist two additional criteria, in relation to the paradigm of absolute autonomy. It follows that system knowledge and resources constitute a factor for autonomy per se. Therefore, increasing both aspects in accordance to the principles contributes to autonomy (*principle of scale*.)

As we have seen —p.76,— there exist two kinds of uncertainty: *qualitative* and *intensive*. We may understand, grossly, that intensive uncertainty is compensated by the *performance* of the system, and qualitative uncertainty by *adaptivity*. We may regard this as an intuitive, general understanding. In this sense, increasing resources would mainly contribute to performance, while increased knowledge would mainly contribute to adaptivity.

We might realize that the qualitative uncertainty may only be compensated for by mechanisms based on general knowledge. Qualitative uncertainty stands for the occurrence of unexpected events, or unknown events which we shall call *qualitative events*. This type of uncertainty requires a dynamic response of the system. More knowledge implies availability of a broader variety of models for explaining qualitative events, and therefore, increased efficacy in adaptivity.

19 (Uncertainty) *Modelling complex systems and environments is affected by* uncertainty. *The systems and environments themselves are subject to it, factor which is amplified by the fact that models are reduced representations. A way of reducing uncertainty–based complexity is to increase knowledge by enhancing it with more models, and also by making them more representative:*

> *[. . .] we add some variables [quantities] to the system [model] [. . .] Each added input contributes, at least potentially, some information that, in turn, reduces the uncertainty regarding the output [. . .] variables. [Kli01, p.163]*

Part III

Perception

Chapter 8

Overview of Studies on Perception

Perception has been studied from many perspectives and by many cultures since ancient times. This has led to multiple descriptions of the phenomenon under specific assumptions and to a number of global views.

The concept of *perception* itself is not clear. Here we shall assume in short that it covers all the way from detecting a certain configuration of the world around the observer to his way of thinking about it. This vision has been adopted by unification of others in which perception refers only to a part of the notion proposed here. Perception has been traditionally distinguished from sensation in some contexts. However there exist more specific and elaborate models [BSZ06], [WWPP06] distinguishing more stages and related phenomena.

We are aware that the notion that will be adopted here is complex: it involves other processes and notions. In this part, we shall try to explain how this notion integrates all aspects.

8.1 The Problem of Perception

A global portrait of perception and the context in which it takes place will show the main ideas, parts and concepts involved. We shall part from a simplified portrait of human vision sketched in figure 8.1[1] as a starting point. This will serve us to develop a unified perspective of the current studies of perception.

There are two contexts affecting the process of perceiving, labelled in the figure as A and B: the first covers the medium in which the perceived object rests within the observer's world; the second, concerns the way in which the world is understood inside the observer.

[1] For an introductory explanation of the biology of human vision, see [KSJ00, p.523-529].

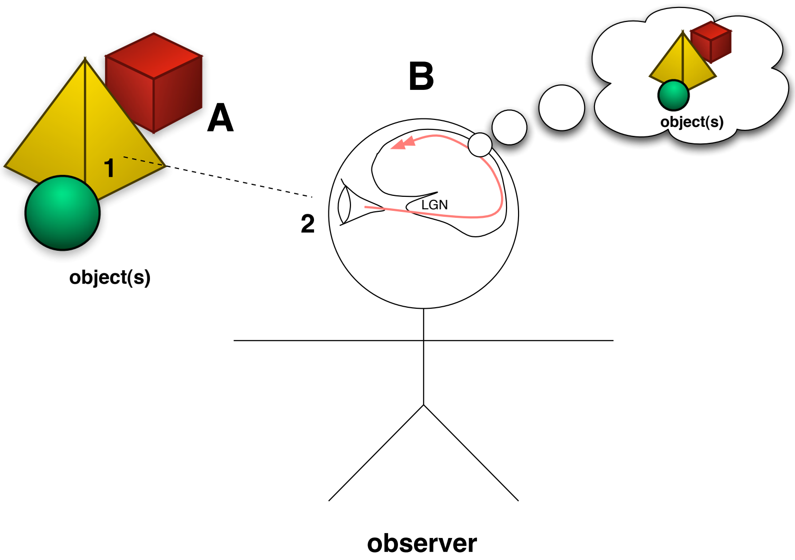


Figure 8.1: Simplified structure of vision in human beings, exemplifying the perception of an object.

The object in the environment is perceived by the observer system because a certain interaction takes place between them. Let us assume that this interaction parts from the object. We shall call it *stimulation*. We shall distinguish the situation in which the stimulation is found at the object by calling it *distal stimulation*, '1' in the figure. The stimulation as found at the receptors in the eye shall be called *proximal stimulation*, labelled '2'.[2]

Apparently, human vision takes place as follows: proximal stimulation is analyzed, filtered and transformed by different parts of the brain including thalamus, superior colliculus, cerebellum, lateral geniculate nucleus and cortex (B). It ends in a certain information relevant to the observer, showing explicitly to him a mapping between the ideas in his mind and the object. It is then when we usually say that the observer has *perceived* the object. We shall call *perceived object* to the ideas in the mind of the observer that refer to the object in the environment.

Perceiving an object depends on two aspects which we shall call *environmental correlation* and *cognitive equivalence*. They are schematically represented in figure 8.2. Grossly, we may say that the distal stimulation is immerse in a medium which yields a corresponding proximal stimulation in the system. This correspondence is the environmental correlation.[3] The system reacts and interprets the proximal stimulation, eventually

[2]This terminology is adopted from [Gog97, p.361], although it is widespread across the scientific community.

[3]We shall assume here that there actually exist objects outside the observer system, and that therefore

considering perceived objects as equivalent to the objects in the environment on some conditions and for certain purposes.

The actual perceived object and the specific way in which it is perceived depend on the sense and context in which the system considers it equal to the object in the environment. We shall call *cognitive equivalence* to the relation of equivalence between the perceived object and the original one.

It must be remarked that this relation is determined by the capacities, purposes and state of the observer system, in the particular context in which it finds itself. Therefore, it may be assumed that different systems, or the same system at different states, would establish a different cognitive equivalence for the same proximal stimulation and object in the environment.

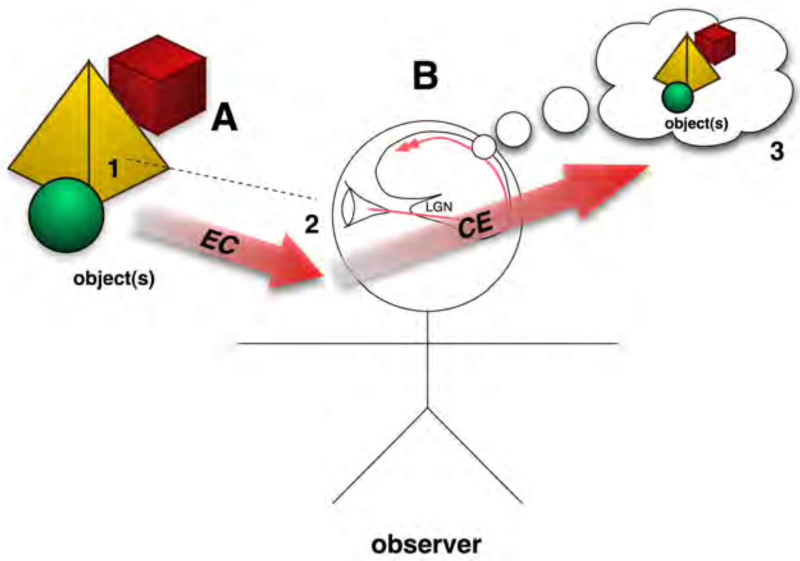


Figure 8.2: *Environmental correlation* and *cognitive equivalence*.

Figure 8.2 and the concepts introduced above serve to provide a first view of the topics and context which studies on perception have addressed. In short, studies on perception have concentrated on cognitive equivalence:

the proximal stimulation does indeed answer to an external, environmental cause. This assumption is actually made —implicitly— by observer systems. However, the existence of an environment and a cognizable environmental correlation have been two classical topics in philosophy, best known by the myth of the cavern —Plato— and the postulate *cogito ergo sum* by Descartes. Although of no principal relevance here, these considerations about the environment and the environmental correlation have a bearing on special cases of perception such as illusion or hallucination. These two phenomena have been used to refute the *Sense-data* theory of perception [Hue], which will be summarized later.

- Exploration and categorization of forms of cognitive equivalence in real systems: rats, primates, humans, etc.
- Studying the biological correlates of cognitive equivalence: biological parts involved and their electrical, electro-chemical behaviours at different levels of aggregation.
- Investigating ways of designing cognitive equivalence relations for artificial systems.
- Extracting general principles and laws from the previous studies: relation of cognitive equivalence with goal-oriented behaviour, survivability, efficiency, resources, etc.

8.2 Major Approaches to Perception

Studying perception through history has yielded a vast knowledge corpus. We are going to develop a unified overview for artificial and biological systems. We shall indicate different references which illustrate significant aspects on particular topics, or which allow developing coherent overviews.[4]

Studies and theories of perception can be classified in three categories, in terms of the part of the process studied being closer to the *proximal stimulation* (2 in figure 8.2) or to the *perceived object* (3 in figure 8.2). We shall review them now.

8.2.1 Near to the proximal stimulation and medium stages

This category of studies focuses on analyzing phenomena, mechanisms and organs for processing of the proximal stimulation.

In biological systems, this includes: (1) Studying the biological structure of sensors and immediate correlates, as well as their electrochemical behaviour in presence of stimuli. (2) Studying the signal-processing performed by the system in relation to the stimuli received.

- [Lev00] and [Sch01] summarize state-of-the-art knowledge about sensation and perception in biological —especially human— systems: visual, auditory, skin perception, taste, smell, the perception of time and perception of movement. They include a catalogue of specific preceptual phenomena (e.g. saccadic movements.) Higher level aspects of perception are briefly treated. There exist larger and more comprehensive sources [KSJ00], [WK01]. The classical, system-oriented analysis in [Arb72] is illuminating.

[4]It has not been intended to provide an exhaustive or comprehensive survey of the literature. The dimension of this task would place it out of the scope of this text. However, key aspects and references have been introduced.

- [Win67] is a classical text on auditory perception in humans. It introduces the basic properties of human voice and musical instruments (e.g. *formants*) and offers an insight into the way they are perceived.
- [Hug01] offers a survey of exotic senses which can be found in animal life, such as biosonar, electroreception, and specific cases of chemical perception.
- Examples of state-of-the-art research can be found in the following sources: [OJSK05] [ZLFK05] investigate medium-scale cerebral reactions to sensory inputs and substratal correlates of sensory functions. [BU06] and [KS05] investigate recognition of different aspects of auditory signals and neural correlates. [Far00] offers an overview of neural correlates across the different low and medium level perception processes.

For artificial systems, correct processing of proximal stimulation may be critical. In fact, most algorithms and artefacts for artificial perception fall within this category. In artificial systems, this part of perception is normally called *preprocessing*. It consists in either segmentation, filtering or combined techniques applied to signals and images: selection and extraction of regions from a larger image, elimination of noise, Fourier analysis and/or transformation, equalization, and edge-extraction among other.

- [Dav97] offers a survey of all major techniques and methods used in artificial vision, from low- and medium-level processing —this category— to object recognition. [GW92] is a well-known reference on digital image processing techniques.
- [Ley92] constitutes a new perspective on visual analysis based on the interpretation of deviations from symmetrical relations within the image. It briefly assesses the analysis of artworks.[5]
- [McD95] proposes artificial methods for rhythm perception and generation based on principles of rhythm perception in humans.
- Two systems discussed in this text, [ASM+06] and [MAHP03], constitute examples of artificial systems based on low-level vision.

8.2.2 Near to the perceived object

These studies concentrate on the process of establishing the cognitive equivalence in systems: the different aspects related with the perceived object.

In summary, there are two main lines in biological studies: (1) Dependence of object recognition with the rest of the system (parts and state: memory, emotions, knowledge, etc.) (2) Features, deduced in previous stages or present in the proximal stimulation, which determine the perceived object.

[5]The Gestalt psychology, introduced in section 8.2.3, conceives perception as the processing of analyzing proximal stimulation for optimal structural criteria. Among these criteria is symmetry. [Ley92] could be regarded as a modern computational approach in this line.

- [BML^{+}06], [BLMLS05], [Man04] [SS04], [SJS03] are studies related to the process of concept formation and recognition in humans. In particular they investigate the creation of concepts and selection of visual features for recognition. [WBS92] offers a wide catalogue of visual phenomena in relation with cues and visual illusions.

- [KS05] proposes a model of music perception rooted in neurophysiology, and points out the influence in musical perception of working-memory and long-term memory as well as other system parts and capacities. [VKWL06] investigates the influence of cross-modal interaction in musical perception. [PJ01] investigates the identification of musical performances in different contexts —varying the patterns of sensory inputs—.

- As we shall see later in the text, memory retrieval and perception (specifically *metaperception*, see chapter 10) are intimately related. This topic is analyzed in [BL03], [KGESss], [RC06] for dependence on emotions, on the degree to which concepts are integrated and to the context in which they are remembered.

- It is particularly significant to this work the notion of *perceptual symbol system*, as introduced in [Bar99]. Against the usual conception of perception as a biological process of recording concepts in form of symbols,[6] this study postulates that the actual recording is schematic. The full meaning of a concept is obtained by a specific *simulator* iterpreting the corresponding shematic representation in a specific context. This develops the point of duality representation/processing indicated in [Mar82].

  It is worth mentioning that perceptual symbol systems as introduced in [Bar99] provide an overall explanation for multiple neurophisiological phenomena observed when perceiving, such as activity in sensorimotor areas (observed in music perception for instance [KS05], [RK03].) Embodied cognition >*section 8.2.3* analyzes cognition from a systemic perspective, addressing this among other topics.

In artificial systems, development is focused on investigating and designing characteristic features, artefacts and methods which allow robust object recognition. The main difficulties are the uncertainty and variability of the proximal stimulation associated to a same object, in particular potential excess/lack of information, and absence of a defined informational structure.

1. Artificial neural networks are commonly used in artificial vision. A classical text on the field is [DhS01]. It describes major types of networks and problems for pattern recognition. Pattern recognition in many networks is dependent on the position, angle and rotation of the pattern in the image. The classical Cognitron and Neocognitron networks [Fuk75], [Fuk80], inspired in biological mechanisms of vision, are significant examples of complex networks which carry out scale-rotation-invariant recognition. [Ull96] provides a comprehensive exposition of problems, methods and approaches to object recognition.

[6] This refers to *amodal* symbols, i.e.: which are independent from the way or *mode* in which they are interpreted.

2. COGENT is described as a *Cognitive Agent to Amplify Human Perception and Cognition* [DG00]. The system implements the whole perceptive process from proximal stimulation to object recognition. In particular, it uses cognitive equivalence relations called *filters* —$2\frac{1}{2}$-D model in [Mar82], >*sec. 8.2.3*— to conceptually interpret analyses of proximal stimulation, called *events* —*primal sketch* level in [Mar82]— (the concepts are a range of possible degrees of emergence.)

3. There exists a line of research focused on perception of complex objects in uncontrolled environments. In automotion, an example of this line is pedestrian recognition in real driving scenarios [BBC$^+$03], [Bro99].

4. Affective computing is a current line of research concentrated on emotional artificial machines which recognize —even experience— emotional states. An overview of the field can be found in [Pic98] [SGQ05]. [Uni05] includes a representative collection of current advances in multiple aspects of affective computing: synthetic emotions, consciousness, etc.

   Other examples of developments in this field are [POMS05], in which emotions are used to increase the accuracy of a reasoning system, and [BD05], in which speech is analyzed for discriminating among 6 emotional states of the speaker.

5. Examples of development in abstraction and grounding of concepts are: [Roy05] experiment of grounding words in perception and action. Also in the line of word-grounding and word computing is [Zad02]. [Cha04] proposes a robot architecture which integrates a model of cognitive development (in the line of the computational approach to perception introduced later.)

8.2.3 Global approaches to perception

Perception is studied relatively to the system and systemic aspects:

- The role of perception within the system. Relevance for: autonomy, goal-oriented behaviour, performance, etc.
- Causes for perceptual phenomena. Systemic explanations accounting perception and also for illusions, hallucinations, particular aspects of perception (perception of movement, perception of volumes, etc.)
- Cognitive (neuro–) science: Relation of concepts, concept formation and concept recognition with neurophysiological substrate.

These approaches have been based in the study of biological systems, especially of the human being. They have been developed mainly in psychology, although many attempt to postulate universal principles of perception. It is worth mentioning that most of the studies in artificial perception are concentrated on specific problems, included in the previous categories of studies. Two examples of general approaches derived from artificial systems, however, will be commented now: [Mar82] and [Sha05].

The two main trends in global studies of perception are *direct perception* and *mediated perception*. Their main postulates are summarized:

Direct Perception

Also called *ecological perception*. It is proposed by J. J. Gibson in two major works [Gib66] [Gib87]. Its main thesis can be summarized by the following quote [Gib87, p.127]:

> The composition and layout of surfaces *constitute* what they afford [the system]. If so, to perceive the system is to perceive what they afford. This is a radical hypothesis, for it implies that the "values" and "meanings" of things in the environment can be directly perceived. [...]
>
> The *affordances* of the environment are what it *offers* to the animal, what it *provides* or *furnishes*, either for good or ill. [...]
>
> If a terrestrial surface is nearly horizontal (instead of convex or concave), and sufficiently extended (relative to the size of the animal) and if its substance is rigid (relative to the weight of the animal), then the surface *affords support*.

Direct perception sustains that animals directly perceive meanings in the environment —in the sense of affordances—, without inference or symbolic processing, and independently from past experiences. In [Gib65], Gibson distinguishes between *direct-* or *first-hand perception* and *mediated-* or *second-hand perception*. It is implied that direct perception corresponds to the real mechanisms of thought of animals and to some forms of expression. Mediated perception would be strictly conventional and limited to specific forms of communication —socially–, culturally–, individually-dependent—.

[Gib66] and [Gib87] explain key aspects of perception as proprioception, perception of movement and visual awareness in terms of direct perception.

Mediated Perception

In some contexts, it is referred to as the *constructivist* or *computational approach*, or as *indirect perception*.[7]

Mediated perception covers a broad range of lines of research. According to mediated perception, perception is conceived as information processing. It would be determined not only by proximal stimulation (as in the *ecological theory*) but also by the knowledge of the system, its state and its capacities. The result of perception would therefore be *mediated* by the system that perceives.

Some major lines within mediated perception are introduced following.[8]

Helmholtz Inferential Approach. The first conception of mediated perception is attributed to Helmholtz [vH05]. It is llustrated as follows [Roc97, Foreword by Stephen E. Palmer, p.xiii]:

[7] *Indirect perception* is also used to refer to the line indicated below as *inferential approach*, represented by Irving Rock among others.

[8] They are not necessarily disjunct, and may actually present points of convergence.

> The observer somehow *adds information* from internal sources, what in modern parlance would be called "heuristic assumptions." In essence, the inferential approach hypothesizes that observers make very rapid and unconscious inferences based jointly on optical information stored in their retinal images and internally stored knowledge of the likelihood of various real-world situations given particular kinds of image structure.

In summary, Helmoltz's view —also called *inferential*— claims that proximal stimulation is analyzed by perception for the most likely interpretation. Criteria of likelihood are assumed to be given by past experience.

Umwelt —Jakob von Uexküll. The concept of *umwelt*, was introduced by Jakob von Uexküll [vU82]. It has been translated as *semiosphere* and *subjective environment*. Usually, the term 'environment' is used in a neutral way, as a photograph. The notion of umwelt refers to the actual subjective environment which an observer perceives; in which sizes, proportions, smells, and all other aspects of the surroundings are enhanced or reduced according to his tastes, needs and desires.

It expresses how the perception of detail and intensity and of the properties of the objects that surround the observer is altered by his actual subjectivity. For deeper insight into the concept of umwelt see [Dee01], [Has03], [Lot02], [Sha]. For reflections regarding umwelt in artificial systems see [Emm01].

Gestalt School of Psychology. The Gestalt movement [Kof63], [Köh69], [WD04] postulates that the perceived object resulting from a process of perception is not the *most likely* (as in the inferential approach,) but the *best* interpretation of the proximal stimulation. This was established in terms of the structural simplicity, symmetry and regularity of the proximal stimulation.

Perhaps the main idea introduced by Gestalt is the emphasis in analyzing the proximal stimulation for its structure, i.e.: the relations between its parts.

Sense-Data Theory. Sense-data theory [Cra] [Fir50] [Hue] propose a model with some analogies to direct perception, based on *sense-data*.

[Hue] analyzes sense-data in comparison to other theories of perception. In summary, sense data are "the (alleged) mind-dependent objects that we are directly aware of in perception, and that have the properties they actually appear to have."

The analogy with ecological perception is the direct character of sense-data, as that of affordances.

However, there are significant points of divergence: (1) direct perception assumes "we are directly aware (only) of things in the physical world," in opposition to the mind-dependent character of sense-data, (2) Affordances —which determine *what* can actually be perceived— are referred to the system "they have to be measured *relative to the animal*" [Gib87, p.127]. This condition is not assumed by sense-data theory.

According to sense-data, what are perceived are the properties *of the object*, not what they could afford the system. Sense–data theory can be regarded as an intermediate theory between direct perception and the rest of trends in mediated perception.

Marr Theory of Vision. [Mar82] is a classical text analyzing vision. It has raised multiple issues of significant relevance to perception. For a synopsis, consult [Mar82, p.329-332]:

- Representation and processing have a dual nature, and are mutually dependent in systems. This is in direct relation with the *perceptual symbol system* approach mentioned [Bar99], and with the *embodied cognition* approach, as well as with most of the other modern branches of mediated perception.
- Vision cannot be explained at a single level of abstraction. Explaining the phenomenon requires studying three aspects of a whole: the computational, algorithm and implementation levels.
- There exist three types of representations involved in vision according to their level of abstraction and perspective: (1) the *primal sketch*, concerned with analyzing the structure of the proximal stimulation for structure and intrinsic characteristics —as in Gestalt— (2) the $2\frac{1}{2}$-D model, in which objects are represented relative to the observer, and (3) the 3-D model, in which objects are represented neutrally, independently from the observer.

It must be noted that these issues are in relation with many key, unsolved aspects of perception and representation: multiresolution perception, context-dependence in knowledge representation, relation between representation and action, multilevel integration and fusion among other.

Indirect Perception, Inferential Approach. Irvin Rock among other researchers propose the *unconscious inference* approach to perception [Fer05], [Fis01], [Hoc74], [Pyl99], [Roc83], [Roc85] [Roc97], [Ull80] ([Pei58] is a classical reference preceding these sources.) It is deeply rooted in Helmholtz's original view. Differently from it, perception is viewed as a purely problem-solving process, in which many factors intervene apart from past experience.

This theory is grounded on experimental evidence showing the importance of *phenomenal* perception against the purely sensory. In other words, showing how the retinal images may differ from the actual perception of a scene due to mediation of inferential processing.

An important notion related with indirect perception is the *percept-percept coupling* —introduced in [Hoc74]—. It refers to links between percepts in a similar way as links between concepts in a semantic network. This shows a way in which past experiences may influence new perceptive processes. The notion will underlie the present work, especially the notion of *referent* and the different mentions to knowledge.

[Sha05] provides a perspective on perception unifying multiple key aspects: incompleteness, top-down and bottom-up information flows in perception, and sensor fusion. Although the text is oriented to artificial systems, it significantly accounts for perceptive phenomena in general. This work, as it will be further developed in chapter 15, is closely related to the fundamental ideas of this thesis.

Embodied Cognition. It is also referred to as *situated cognition*. Sometimes partial aspects of the discipline are referred to as *situated action* or *situated perception*. The term 'situated' denotes that an entity or phenomenon "cannot be studied, described or otherwise fully understood, in isolation from some larger context" [Sto].

Embodied Cognition can be regarded as the current trend in cognitive science. In short, it conceives perception within a network of interrelated processes in the mind, not clearly separable from some stages of action and inferential processes, or explained regardless the substrate —body— in which the processes take place.

The appearing of neurophysiological evidence showing that perceptive tasks involve motor areas has altered the traditional view of separate processes. This issue has been globally studied in terms of perception [Hur01], [Noë04]. However, perception also influences and is influenced by other parts and processes in the system apart from motor areas, as it was mentioned previously [GL05], [KS05], [RK03]. These facts support the inseparability of perception from the rest of the system as a whole, and from this and its environment. In other words, it is accepted that our interpretation of our senses —i.e.: the specific cognitive equivalence we establish— may vary according to our emotions, actions, bodies, etc. This leads to *embodied cognition*, term which stresses the relation between the process and the body in which it is grounded. An overview of the field can be obtained from [And03a] [Chr03] [And03b], [CW04], [SZ01]. An interesting analysis from the point of view of artificial systems from [Flo03]. An interesting architecture, based on the mechanism of *mirror neurons* [Wie06] exploits a world model integrating motor, cognitive and perceptive capabilities. [HBB$^+$06] is an example of applied research on situated cognition for miliatry purposes.

Embodied cognition faces several critical issues which must be explained —the purpose of part II of this work is precisely to illustrate a view on these topics—:

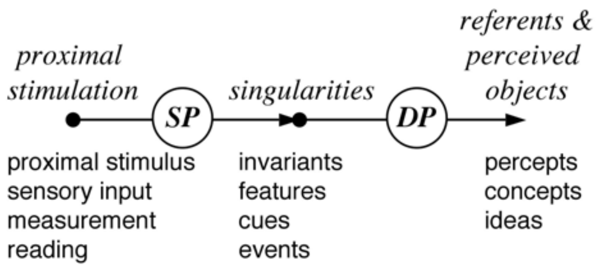
- Integrated action of the system: Coherence, constraints and dependences between perceptive and other processes, organs and elements of a system. Principles underlying system cohesion.
- Embodiment: Dependence between cognition and structural and substratal aspects of the system: capacities, restrictions, features derived from the body -or the *implementation* in artificial systems—.
- Functionality: Relation between cognition and the objectives and purposes of the system. Relation with major aspects of the system as homeostasis and survivability.

Chapter 9

Thesis

The main thesis of this work will be introduced in five points. The next chapters of this part will provide further detail and develop the framework in which these points are to be understood.

I. On the perceptive process. Any perceptive process is based on three aspects: *proximal stimulation*, *singularities* and *objects*. The process consists in relating the three aspects. It is what is called *perceiving* or *perception*. Globally, perception always follows a sequence of two phases which will be called *fundamental sequence* >p.178, represented schematically in the diagram:



SP: Sensory processing / proximal information processing

DP: Directed processing / cognitive information processing

Terminology of this work is written in italics. Alternative terminology found in the literature is written in sans-serif.

SP and *DP* represent the two phases of the fundamental sequence.

The perceptive process is directed to recognizing certain entities in the environment, while ignoring others. These entities to which perception is referred to, shall be called *referents* of the process.

As we were saying, referents are concepts in the system, *objects*,[1] which perception will strive to find in the environment. They are conceptual and cannot be found as such in the environment.

If they actually exist, they will appear in a specific form and body. In other words, as a particular *instantiation* of the actual referents. It is this instantiation which will be represented by perception as a *perceived object*.

The perceptive process might be implicit or explicitly oriented towards its referents. If the process manipulates symbolic representations of its referents, the orientarion will be *explicit*; otherwise it will be *implicit*.

Singularities are patterns in the values of the proximal stimulation.[2] These patterns are attributed by DP to a certain configuration of the objects in the environment. It is this configuration which is represented into the *perceived object*.

This *attribution* consists in actually assigning an equivalence between a state of the referent and the state of the object in the environment. As we just mentioned, a perceived object is therefore a representation of a particular state of a referent which is recognized in the environment: an *instantiated referent*.

II. On the context of perception. Perception is dependent on the system and its environment. Perception is influenced by the rest of the processes in the system and influences the rest of the system in two ways: through the potential explicit effect of the perceived objects, and by inducing changes in the system during the process.

The concepts about systems introduced in part II develop the framework in which perception takes place: multiple perceptive processes grounded in resources which may be mutually dependent or shared, correspondence between perceptive processes, system behaviour, system organization and system objectives and other, which largely determine the purpose, task, capacties and relevance of a specific perceptive process in the system.

III. Cognitive relevance. This point only stresses the fact that perception is referred to concepts which we have been called *referents* of the process: ideas, abstract concepts, objects.

These referents establish the point of view of perception: what is interesting and what is not. In other words, they establish the *finality* of the perceptive process.

The *perceived objects* which result from perception are needed for solving problems, planning actions and monitoring the state of the system and the environment. They are the link between the real world and the operation of the system.

Operations with referents and perceived objects[3] are needed to simulate hypothetical scenarios, to refine algorithms and processes and to enhance knowledge. They are needed to create new referents which can in turn be perceived.

[1] *Object* in the sense of idea, concept, conceptual entity.

[2] They generalize a series of concepts of different fields of knowledge >*comment on p.189*.

[3] These operations are, mainly: generalization, analogy, association, and particularization of concepts.

IV. Perceived domain. A perceptive process perceives over a part of the universe which we shall call *perceptive environment*. It includes the outside of the observer system, *system environment*, as well as —in the general case— parts of the system itself >*introduction to chapter 10 and fig. 10.2*. In other words, this means that perception can recognize referents outside and inside the system.[4]

Perceiving externally or internally to the system —or both— is irrelevant as to the structure and nature of a perceptive process.

There may appear differences as to the grounding, level of processing or other aspects specific to a process and a system. For example, processes which perceive inside biological systems frequently operate upon richer proximal stimulation (essentially in number and nature of inputs,) given the density of nervous/chemical connections inside the system.

V. On interaction between sensory and directed processing. Both types of processing can interact and be mutually influenced throughout a perceptive process. This will be explained and developed in section 10.3.

These points constitute the main ideas underlying this work. They have been listed here in a context–independent form. Detailed comments as to their implications in relation to current conceptions of perception are developed in chapter 15.

[4]Perception inside the system, as we shall analyze in section 10.2 and develop further in later chapters, gives rise to *proprioception* and *metaperception* (among many other phenomena.) Note that these types of perception are in close connection with the principles of the perceptual symbol theory [Bar99] commented earlier.

Chapter 10

Perception

In this chapter, we are going to build an overview of perception. We shall introduce basic concepts defining the entities involved in the process, and the dynamics of the process itself. These concepts will lead to explaining a view of perception in the context of a system. A detailed analysis assessing the informational dimension of the process will be developed in later chapters.

Let us build a basic framework of entities involved in perception by parting from the sketch of figure 8.1. First, let us generalize the human observer into an entity which may be human, biological or artificial. We shall call it *system* or *observer system* depending on the context. We may realize that the observer system covers part B of the process as sketched in figure 8.1, ranging from proximal stimulation to the final, perceived concept.

Around the system there exists a world which we shall call *system environment*. In the diagram of figure 10.1, the *observer system* is perceiving entities in the *system environment*. This is the intuitive notion of perception: a system perceiving over its environment. We shall see in the following sections, however, that it is not actually the whole system that perceives, and not only the system environment the subject of perception. A system —for example humans— perceives parts of itself as well as its environment, and not all of it is dedicated to perceiving; parts of it perceive while other parts are devoted to different tasks. Also, we shall see that the result of perception is more than concepts. Perception derives in concepts, but will also yield other responses that we shall see.

In spite of the lack of accuracy, the intuitive notion of figure 10.1 must be reminded. A system perceives over its environment.

As we have just mentioned, it is only a part of the observer system that actually *perceives*, while the rest performs other functions such as motion control or deliberation. We can imagine the perceiving part as a specialized entity, which we shall call *perceptor*, abbreviated by PR. Let us regard this as a conceptualization. In real systems, especially biological ones, sometimes the division between the functions of system parts is so fine–grained, and the functions themselves so interleaved, that they are difficult to separate.

In fact, a perceptor should not be conceived in isolation, although it will help us

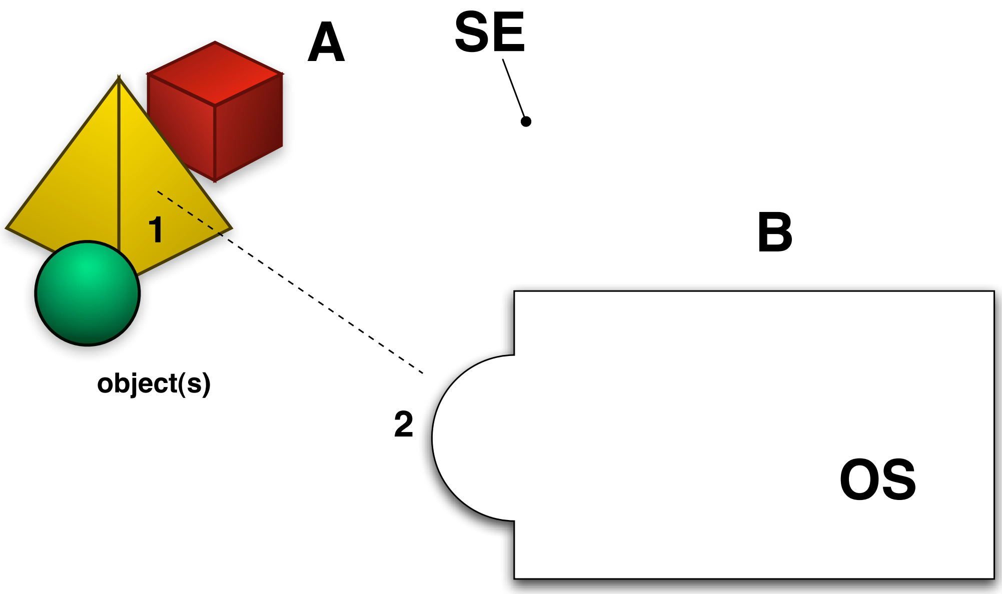


Figure 10.1: Generalization of a human observer into an observer system and the rest of the world, that constitutes the system environment, SE.

now to explain the process. Perceptors of complex systems may be tightly dependent of other parts, and even the processes of perception they execute may require other processes carried out by other parts of the system. The existence of these dependences and couplings only shows that perception happens in the context of a system: its objectives, its state, its capacities...

Dependences may influence the way in which a perceptor operates, and consequently, the whole of the perceptive process. If we return to the example of human vision, we may realize that the number of factors in the rest of the system that may influence the process is huge. Blood pressure, hormonal composition, muscular tone, stress, and all sorts of psychological factors (past experiences, atmosphere, depression, euphoria, etc.)

In order to complete the basic framework of perception, let us separate the *perceptor* from the rest of the *observer system*, assuming the distinction is conceptual. We shall call *perceptive environment* to all which is not perceptor. We may observe that this includes part of the observer system as well as the system environment.

The resulting framework is shown in figure 10.2. In terms of these framework, *perception* is executed by the *perceptor* of the *observer system* over the *perceptive environment*. It is the *perceptive environment* which is actually principal in perception, while the *system environment* of our intuitive notion will appear only secondarily.

In the following sections we are going to analyze the relations between the *perceptor*, the rest of the *observer system* and the *environment* mainly from a structural point of view.

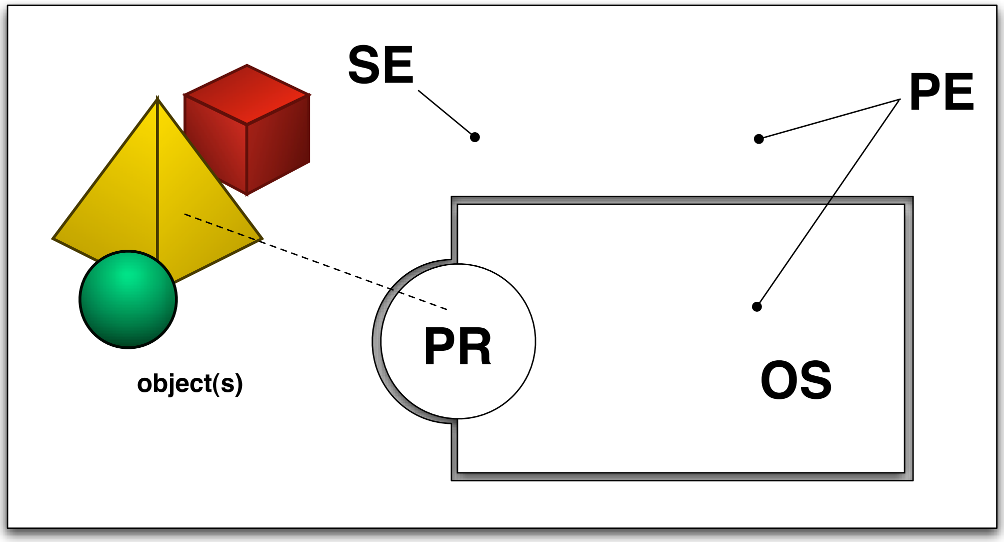


Figure 10.2: General concepts of observer system (OS), perceptor (PR), system environment (SE) and perceptive environment (PE).

The next chapter will cover perception from a *cognitive* perspective.

10.1 Overview of Perception

In the light of the concepts that have been introduced previously, *perception* can be conceived as a process which produces changes in the *observer system* related non-randomly to the state of the *perceptive environment*. In fact, the changes resulting from perception have a meaning related to the system *activity*, as follows from the discussion on system *finality*, of part II. This meaning is a representation of the state of the universe, *relative to the system*. We may realize that the quantities which form this representation are *conceptual quantities* in the system, because they refer —and conceptually stand for— *real quantities* of the environment. Accordingly, perception answers to the basic scheme:

$$\text{proximal stimulus} \xrightarrow{\quad perception \quad} \text{concept}$$

Let us briefly return to the notion of *perceptor* introduced previously. The perceptor can be analyzed as a set of quantities. If we analyze the process of perception in terms of the quantities of the perceptor, we can see that it will take place in three phases:

1. First, the *environment* will induce certain changes in a part of the *perceptor*.
2. Second, these changes will trigger others inside PR, following the dependences among its quantities.

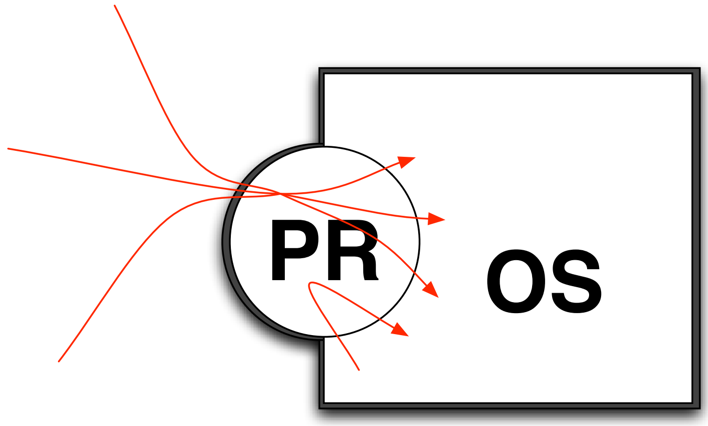


Figure 10.3: Perception. Perception can be conceived as a process which produces changes in the observer system related non-randomly to the state of the perceptive environment.

3. Third, the dependences between the perceptor and the rest of the OS will lead to alterations in the *observer system*.

As we have seen, the alterations generated during the third phase form the conceptual representation resulting from the process. We may infer that the same observer system in a different environment would lead to a different succession of changes; the same would happen if a different system had been in the same environment.

20 (Subjectivity) *If we consider ourselves as* observer systems*, we shall see that our perception of the environment might differ from that of our neighbour. Sometimes, we focus our attention in aspects of reality which are irrelevant for other persons; sometimes we perceive details which are unseen for the rest. Sometimes we meet people who are specifically trained to perceive the environment in a different way.*

Even when remaining to ourselves, we might realize that the same street might not seem the same in different days, even when the light and the time are the same, because our thoughts alter the way in which we look at the world. States as sadness, excitement, expectations influence perception. In extreme cases, for example when suffering depression, objects might appear smaller or larger than they would in normal circumstances for the same individual; places might seem dark regardless of the light. States of euphoria might produce the opposite effect.

These everyday, phenomenological experiences reflect that our perceptor and our way of perceiving, as specialized as may be, operate within the constraints and dependences derived from the more complex context of ourselves, the observer system.

The mentioned notion of perception, between *proximal stimulation* and *conceptual representation*, covers, however, only part of the process. The dependence between the perceptor and the rest of the observer system implies that there derive more changes, other than the conceptual represetation. We shall call them *implicit changes*. In our three–phase sheme, these changes are due in the second. The real process can be represented as in figure 10.4.

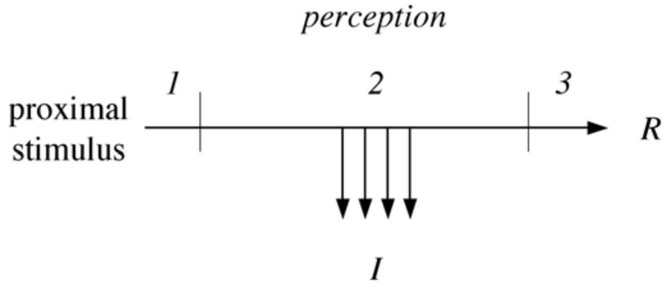


Figure 10.4: Global Process of Perception. Conceptual representation is represented by R. Implicit changes to the system are represented by I. Phases are numbered.

Implicit changes due to perception and those due to the coupling between system and environment influence the operation of the system differently. The perceptive process is subsumed in the function of the system. Therefore, perception is coherent with the *objectives* and the *directiveness* of the system. Both the representation and the implicit changes produced by a process of perception share this directiveness. On the other side, the implicit influence of the environment in system operation through the rest of the coupling system–system environment will, in general, be independent from the objectives of the system.

In summary, we may understand that perception has two values for system operation. Explicitly, it provides a conceptual entity for purposive processes. Implictly, it generates alterations in the organization of the system. Intuitively:

- Informational, the actual explicit knowledge content derived from them (see *encoding* and *representation* as in [New90, p.59].)
- Substratal, affecting the elements of the system and their operation, as in the case of the physiological component of emotions in humans ([Dam00].)

We shall call *percept* to all the changes generated by perception, that is, $I \cup R$. Therefore, a percept has informational and substratal value.

Example 10.1 (Emotions and implicit perception.)
We are aware that many biological systems *have emotions*, even if the definition of what emotions actually are is not clear. We shall adopt here the concept of emotion used in the psychological and neurophysiological disciplines. It is a broad notion, which considers that emotion covers a range of physical, cognitive and social processes in the system

[Dam00], [FMD05], [Gar00], [Pic98]. For a survey of notions of emotions from psychological, technical, philosophical and other points of view, see [Ber06].

[Dam00, p.37] mentions three phenomena in relation with emotions:

- *A state of an emotion:* It is triggered by some perception or process in the system; it is unconscious.
- *A state of a feeling:* It is an unconscious representation of such state.
- *A state of a feeling made conscious:* The conscious representation of the state.

The following table, summarized here after [Gar00, p.75], describes these phenomena in biological terms. They are classified in three categories, which actually happen in approximate sequence.

Level of activity	**Level of emotion (components)**	**Neuroanatomical structures**	**Functions**
Physical	Neurophysiological and biochemical changes	endocrine system, limbic system, hypothalamus, etc.	physiological adaptation, graduation of intensity of the emotion
Mental	Experience (phenomenology)	limbic system, cortical activation, etc.	cognitive level adaptation
Social	Expression	conduct activation, motor cortex, limbic system, hypothalamus, amygdala, etc.	social level adaptation, communication

The physical and first stages of mental phenomena are unconscious and below a conceptual, cognitive level. These processes are triggered by other processes taking place in the system, such as *perception*. They are a reaction of the system architecture to these processes. The purpose of this reaction is listed in the last column, and generically described in [Dam00, p.57]:

> The first function is the production of a specific reaction to the inducing situation. In an animal, for instance, the reaction may be to run or to become inmobile [...] In humans, the reactions are the same, tempered [...] by higher reason and wisdom. The second biological function of emotion is the regulation of the internal state of the organism such that it can be prepared for the specific reation. For example, provide increased blood flow to arteries in the legs [...] in the case of a flight reaction.

In summary, emotions stand for the implicit dynamics of the system. Emotions associated to perception integrate it with the dynamics of the rest of the system. This is called *implicit perception*.

Emotions act at all levels, from the physical level cited above, to the higher cognitive levels. The first, primary stages of perceptive processes may cause emotions of danger or fear, for example, which immediately find a response in the system. Subsequent stages of perception may have their effect in the cognitive processes of the system, as in abductive perception or problem solving [Sha05]. Grossly, we can say that implicit perception triggers adaptation mechanisms and approximate responses to that which is being perceived. We may realize that these actually constitute *mechanisms of directiveness*. An overview of the known roles of emotions in memory, learning and action selection can be found in [CW04]. Specific examples of the role of emotions in music and memory can be consulted in [BL03], [KGESss], [KS05], [RK03], [VKWL06].

There are two major motivations for the current interest in building artificial systems with emotions [Pic98]. First, to achieve socially acceptable/useful behaviour of artefacts which have a high degree of interaction with humans, HCI:[1] robots at workshops or exhibitions, robots for disabled people, advanced HMI.[2] Second, as a means to increase their efficiency in dealing with the uncertainty of the environment. In this particular case, it is envisaged that emotions will improve machine performance for decision-making, action selection, behaviour control, autonomy and confidable behaviour [SGQ05].
■

10.2 Perceptor

In this section we shall use the concepts introduced in the overview of perception in order to investigate the parts of a perceptor. Dynamic issues about the process of perception will inevitably arise from the discussion of the parts of the perceptor. They will be developed in the following section.

We may realize that the couplings between perceptor and system are fundamental for the process of perception. They are, on one side, the path through which the situation of the perceptive environment is input to the system. On the other side, they are the path through which this situation is integrated in the system operation. Our analysis will therefore be based on the couplings of the perceptor through the three phases of perception introduced previously.

During the first phase, the environment induces changes in an element of PR which we shall generically call *sensory system*. We may observe that the quantities which form the sensory system would be mainly independent quantities. These actually determine the proximal stimulus on which perception will be based. Apart from them, a sensory system may include dependent quantities for processes of sensory perception, which will be explained in chapter 11.

During the second phase, the dependences of the sensory system with the rest of PR lead to changes in the values of some of its quantities. The actual values that result

[1]HCI: Human–Computer Interaction.
[2]HMI: Human–Machine Interface.

follow the relations between the quantities, or in other words, the organization of PR.

Let us now return for an instant to the global portrait of perception given by figures 10.1 and 10.2. We may realize that the state of the environment is given to the perceptor by the *distal stimulation* (indicated by '1' in figure 10.1) through the *sensory system*. However, the operation of PR is oriented to the *proximal stimulation* (indicated by '2'). Therefore, the perceptor must provide an equivalence between both.

The calculation of this equivalence is the process of perception. Perception results from the *organization* of the perceptor, therefore from its *structure* and *program* put in a certain system within a particular environment.

The organization of a perceptor, as we shall develop further in the next chapter, is coherent with the *objectives* and *directiveness* of the system. This makes it dependent from the specific system, instant of time and environment, and therefore it cannot be generalized. Some organizational aspects of its *quantities*, however, can be inferred (figure 10.5.)

The quantities of the perceptor may either be internal to PR, or belong to the coupling between PR and PE. In the first case, we shall call these quantities *interdependent*, and denote them by ID.

The quantities of the PR-PE coupling can be categorized. Some of them serve as the input to the perceptive process, and constitute the *sensory system*, SS. These quantities may be coupled to a part of the system itself and/or to the SE. In other words, perception can perceive over the system, the system environment or both.

The conceptual output of the process, the *representation*, will be conveyed to the system by the *representation system*, indicated by RS. They are quantities shared by the perceptor and the system.

However, the perceptor is not independent from the rest of the system or from the system environment. As any system, it cannot be perfectly separated from its environment. Therefore, some of its quantities will be coupled to PE. We shall call them *implicit coupling*. We have used the term 'implicit´ to indicate that it is not within the representational part of the process of perception. We can further categorize the quantities of the implicit coupling, by distinguishing those coupled to the *system environment* from those coupled to the rest of the system. The first shall be called *marginal coupling*. The second, *substratal coupling*. The influence of the marginal and substratal couplings in the process of perception may be reflected *implicitly* in the resulting *representation*.

10.3 Perceptive Dynamics

The previous sections establish the outline of a framework of perception in general systems as to the elements and parts involved in the phenomenon. In this section we are going to analyze the dynamics of perception from a cognitive point of view.

From the point of view of cognition, the activity of a perceptor consists on executing *operations*. They stand for processes with concepts, such as analogy, recognition, association, generalization, etc. At quantity level, these operations stand for operations with

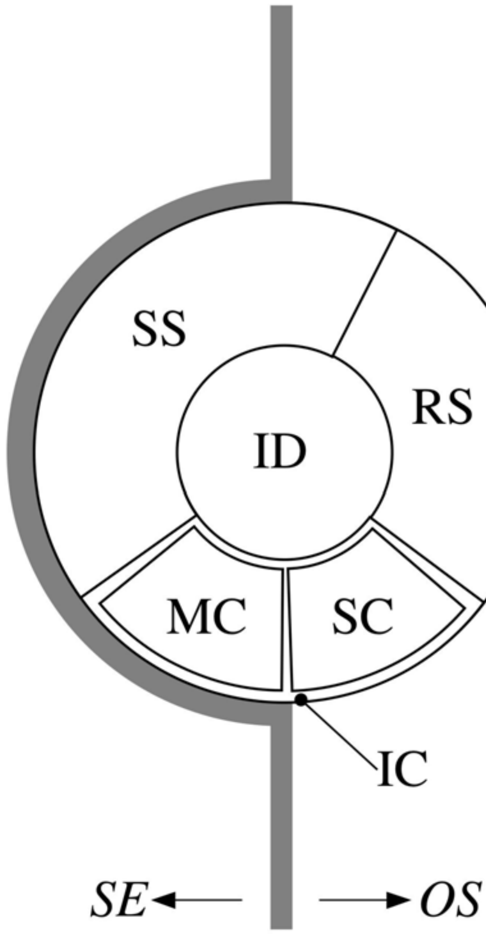


Figure 10.5: Parts of a Perceptor. SS–Sensory system: Inputs to the perceptive process. RS–Representation system: Outputs of the perceptive process. IC–Implicit coupling. MC–Marginal coupling. SC–Substratal coupling. ID–Interdependent quantities.

quantities and quantity-values. As we have seen previously, the operation of a perceptor is based on the values of the independent quantities in the *sensory system*. They are the original state upon which all other operations are carried out by the organization of the perceptor.

Not all processes of perception are equal in terms of abstraction, resources and complexity, and also in terms of the environment in which they take place. This makes that there exists a wide range of possible operations which a perceptor may execute. Operations may cognitively differ, as for example *identification* and *generalization* of objects. Operations may also include subprocesses of non-afferent nature, as in the case of deliberative subprocesses, for example.

As it has been mentioned previously, perception establishes an equivalence between *proximal stimulation* and *distal stimulation*. The perceptive operations and subprocesses leading to this equivalence can be classified in two general categories:

- Subprocesses of acquisition, and of adaptation of the operation of the sensory system, typically by improving its output.
- Subprocesses which transform the output values of the sensory system according to a specific finality.

We shall call the first category of processes *sensory perception* and the second *subjective* or *directed perception*. Ideally, directed perception is based on the output of sensory perception. This defines the *fundamental sequence of perception*:

$$\text{sensory perception} \rightarrow \text{directed perception}$$

The fundamental sequence is a conceptualization of perception in ideal conditions. It will be followed globally in all cases (figure 10.6 (a).) However, as we have mentioned, the specific operations that the perceptor executes may vary greatly. This may lead to different combinations of the perceptive operations and subprocesses: iterations of subprocesses, coordination and combined operation with deliberative and efferent processes, etc.[3] We may consider three main variants of the fundamental sequence, exemplified in figure 10.6.

- Directed perception subprocesses may start before sensory perception is completed, resulting in sensory and directed perception occurring in parallel, case (b) in the figure.
- Directed perception may eventually require further sensory information, so new, intermediate phases of sensory processing may be carried out (*re–sensing*,) case (c). This is an example of perceptive adaptation.
- Perception may involve other kinds of intermediate processes executed by elements out of the perceptor. Typically, complex processes of directed perception may require deductive or abductive phases, case (d).

Perception in a system will result, in general, from many subprocesses of the categories mentioned above, taking place concurrently, sharing intermediate results and resources. The way in which they combine will depend on many factors and may vary in time following the state and finality of the system, and the influence of the environment.

[3] Although the focus of this part is perception, the general framework for autonomous systems of part II underlies this exposition. In particular, it must be remarked that here perception is conceived within a functional structure modelled by nodes and streams.

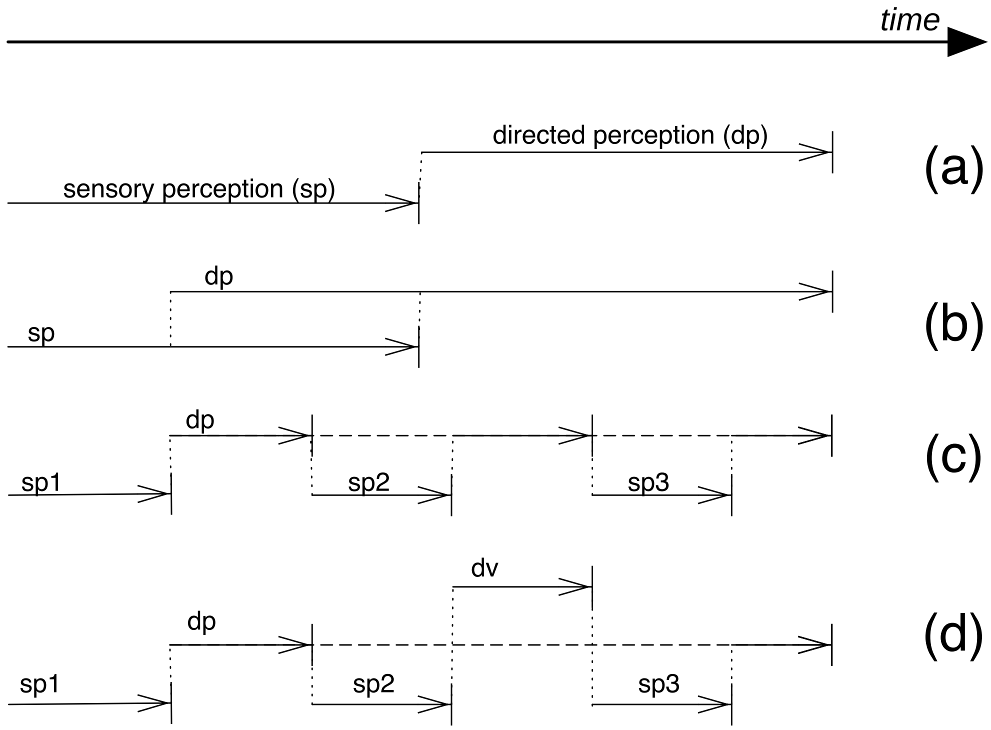


Figure 10.6: Examples of how perceptive processes may occur in time. Case (a) represents the fundamental cycle, and 'dv' stands for 'deliberative processes'.

Example 10.2 (Phases of Perception.)
[KS05] proposes a model of musical perception rooted in neurophysiology. A summary is shown in figure 10.7.

This model exemplifies some of the topics addressed in this text. We may observe that the process of perception is composed by multiple subprocesses. An example of perceptive subprocesses can be found in [BU06]. As we have mentioned, some of these processes may be *deliberative* or *efferent*. Examples of both cases occur in muscial perception:

- "The analysis of musical structure requires the computation of structural relations [...] for example that of the relation between a chord function and a preceding harmonic context [...] Similar operations presumably exist for the processing of rhythm and metre" [KS05, p.579].
- "Certain parts of the action system may be more active when pianists listen to their own performances than when they listen to other pianists play" [RK03, p.608].

[VKWL06] specifically addresses the interaction between different systems in musical performance. The influence of afferent processes in perception is explicitly com-

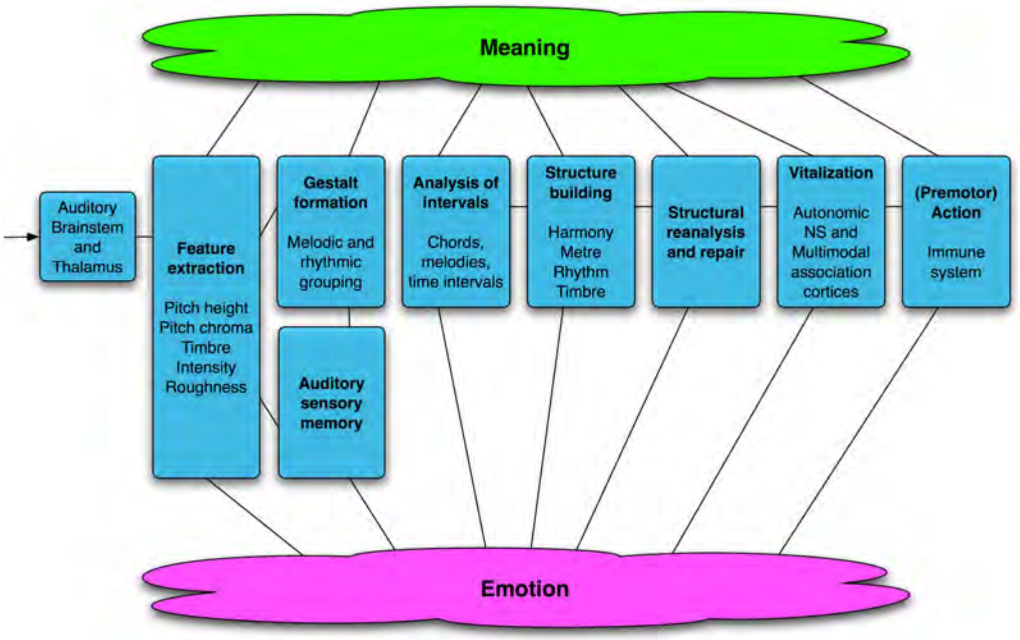


Figure 10.7: Neurocognitive Model of Music Perception, after [KS05, p.579]. Modules to which different aspects of music perception can be assigned.

mented: "musical–equivalence of *paraliguistic gestures* (such as head movements, eyebrow raising, and postural adjustments) convey a significant amount of information that reinforces, augments or augments the auditory signal." This refers to people *hearing* and *seeing* a performance, but it may be concluded from this and the relations between perception and action mechanisms described in [KS05] and [RK03] that this influence of action in perception is also present in the performer: "Recently, it has been shown that music perception can interfere with action planning in musicians, and listening to piano pieces appears to activate (pre)motor activity in pianists" [KS05, p.582].

We may roughly distinguish *sensory perception* and *directed perception* processes of figure 10.7. Sensory perception is composed by feature extraction and the previous stages (brainstem, thalamus, ear.) The rest of processes are directed perception. We may see by the arrow from Structure building to Feature extraction an example of perceptive adaptation as mentioned previously in the text, and shown in figure 10.6 (c).

We must remark the role of Emotion in this model, which, it can be observed in the figure, is linked to all the perceptive subprocesses. Even further, emotions are linked to the rest of the processes in the system (this fact is not part of the model.)

Emotions represent a feedforward and feedback relation between the perceptor and the system through the *substratal coupling*, SC. In feedforward, they progressively convey information obtained by the different perceptive subprocesses to the system. In feedback, the emotions of the system, resulting of perception and of the rest of processes in the system, help *direct* the behaviour of the perceptor.

We must bear in mind that a similar influence of the system environment over the perceptor takes place through the *marginal coupling*. However, this is not directed as in the case of emotions. It can be considered a *perturbance* to the process. ■

10.4 Perceptive Memory

The informational value of perception derives from the *representation*, which is a fundamental part of the *percept*. A representation consists on a specific combination of values of a particular set of quantities. There exist in the system specialized resources which are the substrate for representations. They shall be called, generically, *representation resources*.

The set of *representation resources* associated to the operation of a perceptor is called *perceptive memory*. The informational content represented in it holds the explicit knowledge derived in the corresponding perceptive process. As it will be developed in later chapters, this knowledge is represented under a specific scenario of operation of the perceptor.

Contents of perceptive memory may have different degrees of persistence. Those which are potentially relevant to the system objectives may persist. Otherwise they may disappear, being overriden by future processes of perception.

Persistent contents are included in the *general knowledge* of the system, a factor for system autonomy (introduced in part II.) As we have mentioned, representations are generated by a specific process taking place under specific circumstances. This means that the validity of their informational content is restricted to a particular scenario. New persistent representations must be integrated with the rest of general knowledge in order to enhance their quality (validity, accuracy, temporal value, etc.)

We must briefly remark that the process of *knowledge integration* is highly dependent on *metacognition* and the metacognitive capacities of the system. The effect of knowledge integration on *working memory* (a biological counterpart of *perceptive memory*) is specifically assessed in [RC06]. Related topics are assessed in [BB06], [BL03], [KGESss] and [SSK06].

10.5 Distributed Perception

As we mentioned in part II, the operation of autonomous systems can be modelled in terms of a functional structure that corresponds to a hierarchy of objectives. The functional structure of a system can be expressed in terms of nodes, in which each node is formed by four elements, standing for its afferent, efferent, deliberative and integrative operation and resources.

This model corresponds to a parallel, distributed system. The operation of the system is modelled as multiple processes taking place in different locations of the system concurrently. System cohesion is held both by structural constraints and by unified operation.

Perception in a system stands for a conceptualization of all afferent elements of its functional structure. This means that there is not a unique perceptive process, but a collection of individual processes, each associated to a particular node, and therefore, to a particular objective. Accordingly, the perceptor of the system introduced in previous sections is a conceptualization of a set of concrete perceptors. The afferent element of a node is its perceptor.

Chapter 11

Perceptive Systems

In this chapter we are going to develop a further insight into the concepts of perception. We shall base our exposition in the concept of *perceptive system*, which stands for the perceptor and the associated perceptive memory.

The components of a perceptor are represented in the diagram of figure 11.2. This diagram is equivalent to figure 10.5, but perceptive memory has been represented in order to complete a generic perceptive system. Arrows have also been added in order to show the main flows of changes involved in perception.

11.1 Logic–Grounded Perception

Before analyzing the process according to the perceptive system of figure 11.2, let us develop the overview of the process described in the previous chapter (figure 10.4.) In this overview, the perceptive process takes place between proximal stimulus and a conceptual representation, and implicit changes are derived during the process.

We can say that perception consists on the development of a certain *law of representation* over proximal stimulation, as represented in figure 11.1. The process consists on identifying and characterizing concepts in the environment, relatively to the system. These concepts may stand for real entities or refer to abstract characteristics of them. The process is therefore *referred* to these concepts, upon which the law of representation will operate. These concepts will be called *perceptive referents*.

21 (Referents) *The* referents *are the concepts whose instantiations in the environment are represented by perception. They produce the* distal stimulation *indicated by '1' in figures 10.1 and 10.2.*

Of the concept of 'car,' for example, perception will represent a particular instantiation which may be found in the environment: a silver, large, three–door Rolls–Royce*. It is this instantiation which produces a distal stimulation.*

As was discussed in part II and pointed out in chapter 10, perception is subsumed in the functional structure of an autonomous system. Therefore, the *law of representation* and the *referent*, which define a perceptive process, correspond to the system *objectives*.

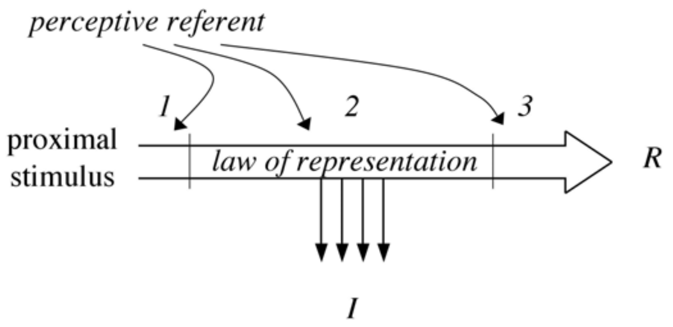


Figure 11.1: Process of Perception.

This implies that the representations and implicit changes generated by perception, represented as R and I in figure 11.1, hold a correspondence to the system objectives.

We have mentioned previously that perception has two values for the system: an *informational* value, which provides explicit representations, and a *substratal* value, which induces changes in the organization of the system. We may consider that these values to the system correspond to two types of perception, which we shall call *logic perception* and *grounded perception*, in relation to the CGSM proposed in chapter 7. *Cognitive* and *grounded perception* stand for different flows of changes in the values of the quantities of the perceptor; they are represented by the long, thin arrows of figure 11.2. The upper arrow stands for *cognitive perception*, which ends in the *perceptive memory* (2). The lower arrow stands for *grounded perception*, which influences the organization of the system through the *substratal coupling* (4).[1]

Cognitive perception is defined by the law of representation, which specifies the process of generation of concepts by the perceptor. The law of representation is based on the perceptive referent, which establishes the actual conceptual space on which the process operates. The perceptive referent is a constitutional aspect of the perceptor (it defines its cognitive operation.) It may affect structural aspects of the perceptor through the substratal coupling (3) and may also be operated upon explicitly by retrieving it from perceptive memory (7). Examples of this are sensory adaptation processes such as the mentioned in figure 10.6 (c), and the relations of auditory sensory memory, working and long-term memories, and knowledge in human music perception [KS05], [RK03].

Grounded perception is given by the substratal coupling of the perceptor. It makes that the state of the perceptor influences the state of the environment (4). We may observe that the dynamics of SC (4), (5), results from the operation of the perceptor following the *law of representation*, and is therefore coherent with the directiveness of the system. On the other hand, the marginal coupling, MC, stands for the non-directed

[1]Obviously, both are relevant to the system, although only the first may be called *cognitive* —or perhaps *explicitly* cognitive, if we accept that all informational processes in the system *are* cognitive—.

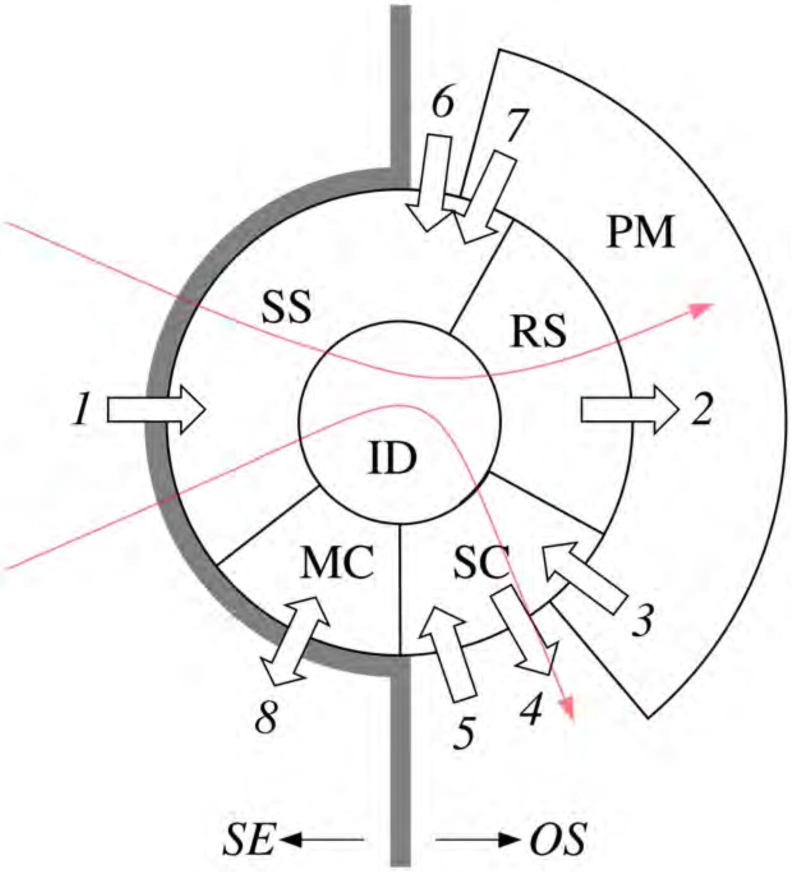


Figure 11.2: Taxonomy of a perceptor. The components are indicated: SS–*sensory system*, RS–*representation system*, SC–*substratal coupling*, MC–*marginal coupling*, ID–*interdependent quantities*. The *perceptive memory* is indicated as PM. The thick line indicates the boundary between *observer system*, OS, and *system environment*, SE. Main flows of information are indicated by numbered arrows: 1,6,7–input, 2–representation, 3–perception referent, 4–substratal percept, 5–operation management, 6–proprioception, 7–metaperception, 8–marginal coupling.

interaction of system environment and perceptor (8).[2]

[2]The influence of the environment on the perceptor through MC is independent from the system, and thus, independent from its directiveness. The influence of the perceptor on the environment through MC can be

22 (Substratal Coupling) *An example of the role of the* substratal coupling *in perception was offered in example 10.2. It is shown how emotions play the role of directing the process of perception according to the global state of the system, and in this way, maintaining the process integrated in the system directiveness.*

Specifically, it was mentioned that the perceptive process has an effect on the system through the substratal coupling, *represented by arrow (4) in figure 11.2. It was also mentioned that the perceptive process, through this coupling, was in turn influenced by the system, indicated by arrows (3) and (5).*

11.2 Perceptive Systems

Some of the interactions of the perceptor have been introduced in the previous section, in order to illustrate logic and grounded perception. Let us explore them in more detail.

Block arrows (1), (6) and (7) stand for inputs to the perceptive process through the sensory system. (1) represents the input from the system environment. This equals the reading of proximal stimulation, in the sense of figures 8.1 and 10.1. The sensory system may also extend to the inside the observer system. This is represented by (6), the sensing over the system itself as if it were an object of the environment.[3] Finally, (7) stands for reading of the concepts in the perceptive memory. These three inputs cover perception over the perceptive environment.

As we have mentioned previously, block arrow 3 stands for conceptual influence of the perceptive memory on the perceptor. This stands for the coupling between the *cognitive system* (see section 7.5.1) and the perceptor. Analogously, block arrow 5 represents the influence of the grounded system on the perceptor. Block arrow 8 represents the interaction between perceptor and system environment.[4]

Processes of perception are based on the components of the figure and the interactions represented by the block arrows. Some perceptors may have only some of the parts, and exhibit only some of the interactions.

Example 11.1 (Referents, quantities.)
Chapter 12 analyzes in detail perception in the DAM —Driver Attention Monitor— system, designed to warn the driver of eventual losses of attention when driving. We shall advance some major features here.

The overall setting is as shown in figure 11.3. A camera is placed on the dashboard of the vehicle facing the driver. It sends frames to an FPGA board installed in the dashboard, which processes the signal and deduces the loss of attention of the driver, emitting an acoustic warning when exceeding a certain threshold.

The system perceives seven *referents*: 1. situation of driver eyes opening or closing, 2. position of the eyes in the image, 3. estimated position in the immediate future,

regarded as a 'side effect' of its operation.

[3]Ultimately deriving in *proprioception*.

[4]Interactions of the substratal coupling have been deaggregated into arrows (3), (4) and (5) in order to address them separately in the text. This has not been considered necessary for the marginal coupling, and only one, bidirectional block arrow, (8), has been used.

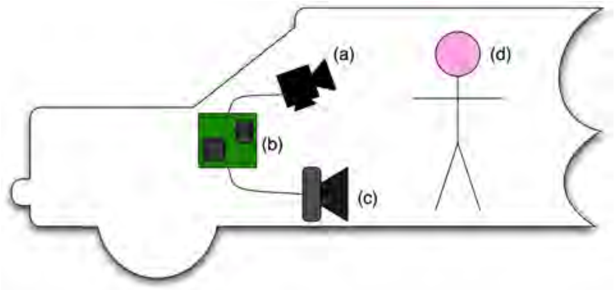


Figure 11.3: Overall Setting of the DAM System in the vehicle. (a) —camera, (b) —FPGA board, (c) —loudspeaker, (d) —driver.

4. short–term environmental light conditions, 5. threshold for discriminating between eyes opening/closing, 6. brightness of the incoming image (instantaneous environmental conditions) and finally, 7. level of non-attention of the driver.

Some of the subprocesses involved in perceiving these referents follow all stages parting from the actual image of the camera and ending with a representation. Others combine the representations generated by other subprocesses and perceive from them. For example, the position of the eyes is perceived from the actual readings of the camera. The level of non-attention is perceived by considering a sequence of representations of referent 1.

The whole system can be modelled as a *node*, in which the *afferent* element (perceptor) is well developed, and the *core* and *efferent* components are elementary.

The *sensory system* is formed by the camera. There exists interaction of type (3) (see figure 11.2, given by two memory registers containing thresholds involved in perception of environmental conditions and level of non-attention. The system also uses memory registers for storing representations. These are the *representation system*. The representation system and the other registers constitute the *perceptive memory* of the system.

Interactions of type (8) of the indicated in figure 11.2 are given by the influence on the performance of the electronics of environmental temperature, quality and continuity of power supply, mechanical tensions and other such dependences with the environment. ■

11.3 Senses

In this section we are going to explore the concept of *sense*, which is a basic notion for explaining the perceptive process. Briefly, we may say that a *sense* is a kind of perceptive process, specialized in observing a particular referent. When referring to this kind of processes, we shall use the term *sense perception*. In example 11.1, we mentioned senses implicitly as 'perceptive subprocesses.'

In usual language we use the word 'sense' to refer to smell, touch, etc. We also use the term to refer to a capacity for identifying certain events by inexplicable factors, which we usually experience as an intuition. We may also refer to a capacity to interpret certain situations developed by training or practice. They are all complex examples of the general notion of *sense* proposed here.

As we have just mentioned, a sense is a specialized perceptive process. As any perceptive process, it is associated to a perceptive referent. We shall call *sense referent* to the perceptive referent of a sense.

In the context of the *functional structure* of the system, the senses form part of the afferent elements of nodes. An *afferent* element of a node may be formed by multiple senses (as we have seen in example 11.1.) On the other hand, the process of perception that the element carries out as a whole —which may include multiple sense subprocesses— has an associated referent, which we shall call *perceptive referent*. It is a superset of its sense referents.

Let us analyze a sense more formally. We may denote the *sense referent* by $\mathbf{V}$. The process of perception carried out by the sense consists in recognizing the referent in the *perceptive environment*. What is really recognized is an instantiation of the concept that the referent represents, not the concept itself. Sense perception is, in other words, generating a representation of the observed instantiation. This consists in a set of quantity values, to which we shall refer as *value of the referent*: ν.

As any perceptive process, sense perception is a composition of sensory and directed subprocesses. Let us denote a particular sense formally by $\mathbf{S}$, and sensory and directed subprocesses by subindices 's' and 'd' respectively, and composition by '$\circ$':

$$\mathbf{S} = \mathbf{S_s} \circ \mathbf{S_d}$$

The operation of a sense takes place over the perceptive environment. Formally, we can express this by:

$$\mathbf{S}(PE) = (\mathbf{S_d} \circ \mathbf{S_s})(PE)$$

Let us develop a brief reflection on $\mathbf{S}(PE)$ by returning to the sketch of the context of perception illustrated in figures 8.1 and 10.1. We may observe that perception represents the objects in the environment.

These objects generate the *distal stimulation* to the process, indicated by '1' in the figures. They are distinguished from the rest of the environment and characterized according to the sense referent. However, the input to sense perception is *proximal stimulation*, indicated by '2.' Therefore, sense perception will construct a representation of '1' parting from '2.'

This will be possible because a correlation holds between '1' and '2' in the environment. Designing a suitable sensory system for sensing '2,' perception might construct '1' through knowledge about the environmental correlation. Knowing '1' certain aspects of it could be extracted, which would allow to identify and characterize an object. We shall call them *singularities*.

23 (Singularities) *The concept of* singularity *generalizes and unifies a series of notions about patterns on which perception of concepts is based, especially in the early stages of the perceptive process. These patterns may be referred to as* regularities, invariants, geons *(for an overview see [Sch01, p.160-166] and [Ull96, p.13-35].) The most widely spread terms are* cue *and* feature*, used for both natural and artificial systems [DhS01], [Lev00], [Sch01], [Sel59], [Ull96]. These terms and their underlying notions are used with a similar, but not equal, meaning.*

In some cases feature *refers to a specific regularity or elementary pattern to be found in a sensory reading, characteristic, typically, of a shape. However, specific relations between such features which are equally characteristic are excluded from the meaning [Sel59]. The term* feature *is also used to refer to regularities at a particular instant of time, while calling regularities in time* events *[DG00]. [BML$^+$06] and [SJS03] explore conceptualization by* cues *in humans, thorugh vision.* cues *are understood as patterns at sensory level —like* features *in the previous senses— but also at a higher level of abstraction, as special arrangements or relations between visual objects, especially occlusion and partial occlusion. Regarding auditory perception, [Win67] describes how* formants *act as* cues *to the recognition of tone character and colour acting.*

The concept of singularity *introduced here is based on the same principles as the existing notions: invariance, characteristic, regularity. The major aspects of generality are (1) that singularities can exist at physical or conceptual level, at any level of abstraction. The previous notions confine singularities to a specific level. (2) That singularities may refer any kind of pattern in the proximal stimulation: regularities in time, characteristic aspects of shape, invariant aspects, properties, events, relations between concepts, etc.*

Point (1) requires some further explanation. Singularities *may exist at any level of abstraction due to the fact that perception can take place at any level of abstraction. As we have seen, the* sensory system *of a perceptor may be coupled both to the system and to the environment. When perceiving over the system, it can perceive by* proprioception*, interaction (6) in figure 11.2, or by* metaperception*, interaction (7). In the case of proprioception, singularities may consist on patterns of physical measurements, such as in the previous examples.*

However, metaperception is an example in which the sensory system will sense referents *already perceived, stored in system memory. Therefore, the* singularities *on which metaperception is based are clearly conceptual, such as relations between representations, size of the representations, number of representations, etc. We might realize that, although the underlying notion of singularity in this case is similar to the ones of* features *mentioned above, the actual kinds of singularities involved would significantly differ.*

Example 11.2 (Singularities in an Artificial System.)
Let us consider a vehicle system to detect road lines, which could eventually be integrated with other systems in order to warn the driver about potential lane departure [ASM$^+$06]. The system is similar in setting to that described in example 11.1, figure 11.3, with the difference that the camera is facing the road instead of the driver. The system *perceives* the road lines by analyzing the frames sent by the camera.

A first stage of image processing (segmentation, filtering) eliminates all information but the edges of the objects in the image. The position of the road line is detected by identifying a certain *pattern* of pixels in the image of the edges. This pattern is the *singularity* upon which the position of the line (*referent*) is inferred.

A difference in concept between singularities and referents may be clearly observed in this example. *Singularities* stand for a rearrangement, or in general, a function of the sensory information. By knowing a singularity, properties of the original sensory reading can be known. Referents, on the other side, exist conceptually, whether they are perceived or not, as in the case that the vehicle should circulate on a small road without lines or a path. They are inferred from the singularities. In this case, the inference is direct. The position of the road line is regarded as the center of mass of the sequence of pixels forming the singularity. However, it could have been defined otherwise (eg. the position of the first pixel of the sequence, the last, the average position of the singularities detected in a sequence of frames.)

In summary, *singularities* are characteristics of the sensory readings, while *referents* are concepts that exist independently. A specific perceptive process may establish a certain relation between both, in order to perceive referents parting from singularities. Eventually, other relations could be established or other singularities considered for perceiving the same referents. ■

The distinction between sensory and directed processing is derived from the fact that *perception*, entails two types of relations. First, the environmental correlation between *proximal stimulation* and *distal stimulation*. We shall call this *environmental correlation*. Second, the equivalence between *distal stimulation* and the *sense referent*, parting from singularities. We shall call this *cognitive equivalence*.[5]

The result of the sense perception process, $\mathbf{S}(PE)$, is really an estimation of the value of the referent instantiation (ν). The difference between the actual value of the instantiation, and the estimation is due to inaccuracies in the solution of the two equivalences.

It must be mentioned that $\mathbf{S}(PE)$ stands for a representation ('R' in figure 11.1,) which is only part of the associated *percept*, as it has been explained. Implicit changes associated to perception (represented as 'I') take place during the generation of $\mathbf{S}(PE)$. It therefore must be understood that $\mathbf{S}(PE)$ reflects implicit changes, although they will not appear explicitly (this was advanced in chapter 10.)

Let us now consider *sense perception* more closely, in order to find out the actual meaning of $\mathbf{S_s}$ and $\mathbf{S_d}$. We may refer to the fundamental sequence of perception, the basic ideal sequence followed by perceptive processes, to model sense perception as:

$$PE \xrightarrow{\mathbf{S_s}} S_s(PE) \xrightarrow{\mathbf{S_d}} S_d(PE) = S(PE)$$

As we have seen, this sequence holds in any perceptive process globally, but alterations may occur in real conditions of operation. We shall consider it in the ideal form for clarity, without loss of generality.

In the light of the discussion above, we may realize the difference between environmental correlation and cognitive equivalence in informational terms. The first is objective, in that it is determined by the environment, and the second is subjective, in that it stands for the particular interpretation of the first by a specific system.

Sense perception generates a solution for the environmental correlation and the cognitive equivalence for the current operating scenario: state of the environment, state of

[5]The term *equivalence* will be used more frequently than 'relation' or 'correlation.' It is used to remark that the *system establishes a relation of equivalence*, not saying that the equivalence really exists in the particular way the system establishes it. Actually, the equivalence is arbitrary.

the system, directiveness. We may associate the solution of the environmental correlation to sensory perception, $\mathbf{S_s}$, and the solution of cognitive equivalence to directed perception, $\mathbf{S_d}$. As we shall see, the solution of the environmental correlation provided by perception is biased for the current scenario of operation.

In order to refer to $\mathbf{S_s}$ and $\mathbf{S_d}$, we shall use the terms *proximal information processing* and *cognitive information processing*, instead of the generic 'sensory' and 'directed'.[6] In these terms, the result of proximal information processing is the set of singularity values. It is generically represented above as $S_s(PE)$. The result of cognitive information processing is the estimation of the state of an instantiation of the sense referent in the environment, represented above as $S_d(PE)$.

11.3.1 Contextual Aspects of Sense Perception

In this section we are going to see how sense perception is related to the rest of the system. This will illustrate the context of the perceptive process in cognitive terms, and explain its subjectivity.

As we have mentioned, sense perception is a specialized perceptive process. It performs a selective analysis of the perceptive environment. In other words, it adopts a particular *point of view*. In particular, the point of view of the system (derived from objectives, organization, directiveness, finality, etc.) More formally, we see that a *point of view* consists in a specification of several points [Kli69, p.39]:[7]

- A resolution level.
- A set of quantities to be considered.
- Time–invariant relations between the quantities.
- The properties which determine these relations.

Bearing in mind these considerations, let us particularize the generic portrait of perception of figure 11.1 for sense perception, as shown in figure 11.4. We may clearly see from the figure that sense perception is defined by the sense referent and the grounded law of representation.

The resolution level, first aspect of a point of view, defines the operation of the sense in time and space. We may consider this mainly as an implementational aspect, therefore regarding the *grounding* of the sense law of representation. It is derived directly from the perceptive resources of the node on one side, and from the resources of the rest

[6]The generic terminology, 'sensory/directed,' is adapted to the system as a whole, and to related concepts such as *directiveness*. The new terminology is better adapted to information processsing.

[7]In other words, analyzing a system implies focusing on some aspects, quantities, which are observed at specific locations and instants of time (resolution level.) These quantities are analyzed in search for relations between them which explain a pattern of behaviour. Finally, the causes for those relations are studied in order to completely understand the system.

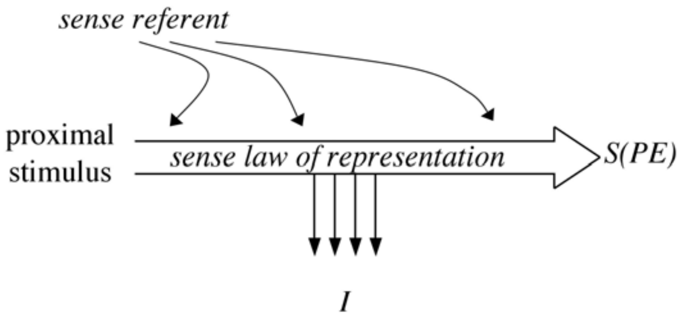


Figure 11.4: Process of Sense Perception.

of the system on the other. Resolution, nevertheless, might eventually have influence over the informational value of the perceptive process.[8]

The sense referent, **V**, on the other hand, is given by the informational aspects of the point of view: the considered set of quantities, time–invariant relations and properties. The process of perception will analyze the perceptive environment relative to them.

Example 11.3 (Point of view.)
Let us return to the system of example 11.2, for road line detection, for analyzing the *point of view* of the system.

Set of quantities: The *sensory system*, namely the CCD camera facing the road, is the actual set of quantities defining the point of view. These are the quanitites whose state is going to be monitored. But not all of the 720×576 pixel array is considered. For reasons explained in [ASM$^+$06] regarding computational load and relevance of information, only a region of interest, RoI, of 102×32 pixels is analyzed.[9] This RoI is situated at the lower right, in order to monitor the road line (see figure 11.5.) The RoI is the set of quantities considered by the system for its analysis of the environment.

Resolution level: The road projects onto the camera depending on the relative angle at which it is oriented, and the optics of the camera itself. In fact, these parameters are subject of adjustment by the designer. A given configuration results in a certain relation of correspondence between distance in the road and pixels: 1 pixel = r metres. r is the space resolution considered by the system (see figure 11.6.)

[8]The resolution level conditions the informational capacity of the sense and vice versa. A high level of resolution, temporal or spatial, implies more data from which to extract richer information. Low resolution may provide insufficient data from which to interpret.

[9]In fact, the system considers three regions, but we shall consider only one for clarity. Considering two more is straightforward.

Figure 11.5: Set of quantities.

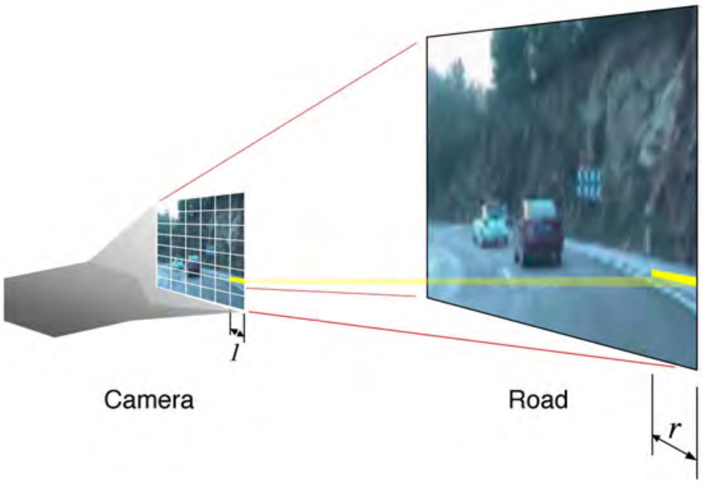


Figure 11.6: Resolution level.

Time resolution is given by the frame rate of the camera. In this system, frames were delivered at PAL rate: 25 fps. Therefore the time resolution is 40 ms.

We should remark that resolution is homogeneous in this system, as in the majority of artificial systems. This might not necessarily be so. For example, regarding human vision, the resolution of the human eye is greater at the fovea, which is 'the part of the human retina which is specialized for detailed vision' [Lev00, p.53].

We should also remark that some systems may change their *point of view* by altering the resolution level. Examples of this are autofocus in artificial systems, and accomodation in vision. Accomodation is 'the mechanism that changes the shape of the lens in order to bring an image to sharp focus on the retina' [Sch01, p.62]. For cognitive, psychological and neurophysiological perspectives on accomodation see [Arb72, p.197], [Gib87, p.217], [vH05, p.123], [Lev00, p.39], [Roc97, p.380], [Sch01, p.62].

Time–invariant relations between quantities: The *singularity* for which the system explores the system stands for a relation between sensory system quantities. It is defined on a filtered, black and white version of the original. The relation is defined for any sequence of 10 consecutive pixels satisfying:

$$b\_b\_w\_b\_b\_b\_b\_w\_b\_b$$

Where pixels are separated by '_' and the letters stand for b—black, w—white. This relation, however, has been simplified for clarity with respect to the system of [ASM^{+}06]. It corresponds to a previous, earlier version developed at INSIA[10] by Prof. Félix Moreno.

Properties: The property that will be deduced as the cause for the considered time–invariant relation is actually the perceived referent: the existence of a road line.

The time–invariant relation explained above yielded robust line detection in real driving conditions. Other time invariant–relations were tested in large series of experiments which produced many false detections (inferring the existence of a line when there was no one present,) typically due to the presence of shadows or stains on the road.

The robustness of the time–invariant relation of this example has made the inference of the road line straightforward. In many cases, especially when searching for properties which are difficult to characterize, the time–invariant relation by itself does not lead to recognizing the desired properties of the environment. In these cases a complex inference process may be necessary. That is precisely the case of a researcher investigating the causes for intrincate phenomena.

■

We may realize that the point of view of sense perception only defines the law of representation partially, through the resolution level. The informational processes for solving environmental correlation and cognitive equivalence are determined by the algorithm of the law. They are parts of the system knowledge, selected and grounded during *functional decomposition*.

In the following sections, proximal and cognitive information processing are going to be described in detail. Sense referent and the kinds of individual processes constituting the law of representation of a sense will be described. The global portrait of sense perception is shown schematically in figure 11.7, and in detail in figure 11.8, in which all elements and subprocesses have been represented.

---

[10] Instituto Universitario de Investigación del Automóvil (UPM) —Automobile Research Institute—. URL: http://www.insia.upm.es

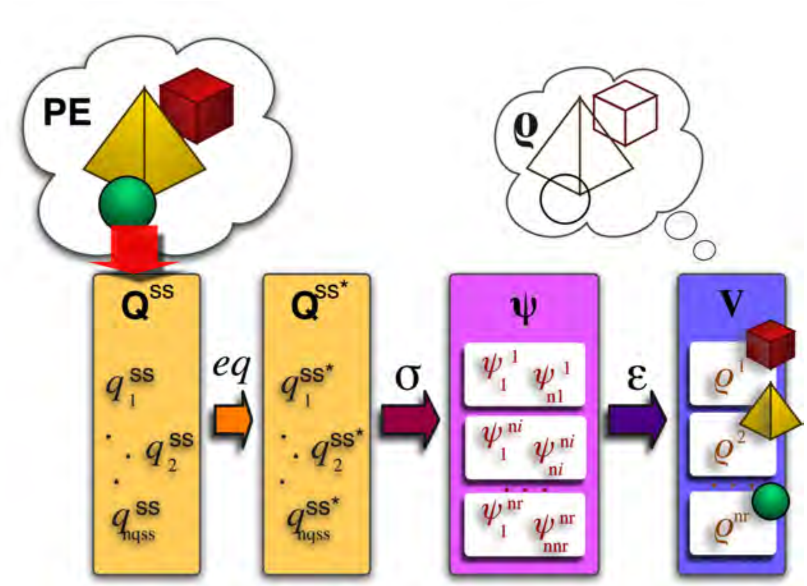


Figure 11.7: Schematic Diagram of Sense Perception. The sequence of processes is shown. 'eq'— *equalization* processes, 'σ'— *singularity equivalence*, 'ϵ'— *cognitive equivalence*. Note that as a result of perception, an estimation of the referent values results, represented by the small coloured shapes. They are instantiations of the actual referents —concepts— indicated by wireframe shapes.

11.3.2 Cognitive information processing

In this section, we are going to analyze cognitive information processing in detail. We shall discuss the cognitive dimension of sense perception, carried out by $\mathbf{S_d}$. Then we shall end by proposing a formalization for cognitive information processing.

In order to analyze the cognitive aspects of sense perception, let us develop a further insight into the sense referent. As we have mentioned, it consists in a series of conceptual entities to be observed in the perceptive environment. Namely: quantities, time–invariant relations between these quantities, and the properties that explain them. We shall refer to quantities, relations and properties generically by *referents*. We may use the following notation:

$$\mathbf{V} = \{\rho^i, \quad i = 1 \dots n_r\}$$

$\mathbf{V}$ stands for the sense referent, ρ^i for referent number i, and n_r for the total number of referents considered in the process.

Now, let us consider sense perception within the functional structure of the system.

As we have mentioned, sense perception is adapted to the operation of a specific node within a functional structure. This means that it corresponds to a certain finality.

The referent and law of representation of a sense are therefore adapted to this context. This means that the sense referent defines a partition of the perceptive environment which is of relevance to the *finality* of the system, and that the law of representation will specify operations which are also consistent with it. In other words, the sense referent fixes which entities can be recognized in the perceptive environment, which relations can be expected to hold between them, and which properties are likely to cause them. It also establishes the relative importance of each against the rest. The law of representation will analyze the environment according to these definitions.

For a specific finality, some referents might be more relevant that others, require more resolution or a deeper analysis than the rest. Therefore, the perceptive process, speaking from a cognitive perspective, does not consider an objective portrait of the environment. It considers a distorted environment in which some parts do not exist, others seem simple, and others are complex and rich of detail. We shall call this notion the *cognitive perceptive region*, and denote it by $\Re$.

We may realize that the cognitive perceptive region holds a certain relation with the perceptive environment, which we can represent as a function, and denote by $\mathcal{F}$, so that:

$$\Re = \mathcal{F}(PE, \mathbf{V})$$

We shall call $\mathcal{F}$ the *umwelt function*.[11] Cognitively, the perceptive region differs from the environment in two main aspects. First, it is *structured*. Second, it reflects the finality of the node. In order to illustrate the cognitive dimension of $\Re$, we may refer to its biological equivalent, the concept of *umwelt*:

> Everything that falls under (...) an Umwelt is altered and reshaped until it has become a useful meaning-carrier; otherwise it is totally neglected.
>
> [vU82, p.31]

Therefore, in cognitive terms, perception equals to observing the state of the *cognitive perceptive region*.

Let us now express the cognitive aspect of sense perception formally, in terms of cognitive information processing. As we have mentioned, cognitive information processing analyzes singularities in order to characterize the sense referent in the perceptive environment.

$$S_s(PE) \xrightarrow{\mathbf{S_d}} S(PE)$$

[11]The original notion was proposed by Jacob von Uexküll referred to living creatures. *Umwelt* is usually translated as 'subjective universe'. For an introduction see [Sha]. Here we have presented a notion for general systems.

Singularities result from proximal information processing. Let us assume that a particular referent, number i, is perceived by analyzing a particular set of singularities. Formally, that referent ρ^i is perceived by analyzing the set of singularities $\Psi^i = \{\psi^i_k, \quad k = 1 \dots n_i\}$. n_i stands for the number of singularities associated to referent i. ψ^i_k represents singularity number k.

Then, cognitive information processing for this referent stands for solving the cognitive equivalence between Ψ^i and ρ^i:

$$\rho^i = \epsilon_i(\Psi^i)$$

11.3.3 Proximal information processing

As we have seen, proximal information processing stands for observing singularities. We have formalized it as:

$$PE \xrightarrow{\mathbf{S_s}} S_s(PE)$$

PE actually stands for the state of the sensory system associated to the sense. In other words, proximal information processing is based on the values of the set of quantities forming the sensory system. We shall denote this set by $\mathbf{Q^{ss}}$, so that:

$$\mathbf{Q^{ss}} = \{q^{ss}_j, \quad j = 1 \dots n_{qss}\}$$

Where q^{ss}_j stands for quantity number j, and n_{qss} for the total number of quantities of the sensory system.

This formalization will serve to explain two kinds of processing of proximal stimulation. The fundamental process, as we have mentioned, consists in the observation of singularities; it stands for solving the *singularity equivalence* for the proximal stimulation. Proximal stimulation processing, however, also covers a different kind of processes which we shall call *equalization*.

Equalization

If we denote the possible domain of variation of a sensory system quantity q^{ss}_j by $\mathbf{D^{ss}_j}$, we may conceive equalization processes as a function eq_j, so that:

$$\begin{aligned} eq_j : \mathbf{D^{ss}_j} &\longrightarrow \mathbf{D^{ss*}_j} \\ q^{ss}_j &\longmapsto q^{ss*}_j \end{aligned}$$

The superscript '$*$' stands for 'equalized'. We see that equalization stands for altering the value of the quantity. There may be two main causes for processes of equalization:

- Optimization of the values of $\mathbf{Q^{ss}}$, by correcting the possible deviations between the proximal stimulation and the observed values. This function is a generalization of error correction in measurements.

- Adaptation of $\mathbf{S_s}$ to the point of view of the sense. This means that proximal information processing is adjusted to cognitive aspects of the sense, for example by introducing aspects of the umwelt function $\mathcal{F}$.

Singularity Equivalence

Let us return to the global formulation of proximal information processing, expressing it in terms of the quantities of the sensory system:

$$PE \quad \rightarrow \quad \mathbf{Q^{ss}} \quad \xrightarrow{\mathbf{S_s}} \quad S_s(PE)$$

Let us develop this formulation at a lower level of aggregation. We may consider the set of singularities associated to referent number i, which has been introduced previously, $\Psi^i = \{\psi^i_k, \quad k = 1 \ldots n_i\}$. Singularity number k, in turn, results from the values of a certain set of sensory system quantities, which we shall indicate by Q^i_k.

The relation between singularity k and the set Q^i_k will be called *singularity equivalence*. It will be indicated by σ^i_k Therefore, formally, for each singularity we have:

$$Q^i_k \quad \xrightarrow{\sigma^i_k} \quad \psi^i_k$$

It holds that $Q^i_k \subset \mathbf{Q^{ss}}$. We may realize that Ψ^i, is calculated upon a set of sensory system quantities, which we may call *referent sensory system*, and denote by Q^i, so that:

$$Q^i = \bigcup_{k=1\ldots n_i} Q^i_k$$

11.4 Representation and Perception

Representation in perception corresponds to block arrow 2 in figure 11.2. We use the term *representation* in three senses:

- To refer to the process of driving a system to a state which is understood to hold a correspondence with the state of another system.

- To refer to the state that results from such process.

- Either to refer globally to all individual processes of representation included in a perceptive process, or to refer globally to all the representations that result from them.

In the context of perception, the process of representation stands for configuring a part of perceptive memory according to a specific relation with the state of a part of the perceptive environment. We have introduced this relation in previous sections with the term *law of representation*.

$$
PE \rightarrow \mathbf{Q^{ss}} \rightarrow \begin{bmatrix} q_1^{ss} & \xrightarrow{eq_1} & q_1^{ss*} \\ q_2^{ss} & \xrightarrow{eq_2} & q_2^{ss*} \\ \vdots & \vdots & \\ q_{nqss}^{ss} & \xrightarrow{eq_{nqss}} & q_{nqss}^{ss*} \end{bmatrix} \rightarrow \begin{bmatrix} Q^{1*} \begin{bmatrix} Q_1^{1*} & \xrightarrow{\sigma_1^1} & \psi_1^1 \\ Q_2^{1*} & \xrightarrow{\sigma_2^1} & \psi_2^1 \\ \vdots & \vdots & \vdots \\ Q_{ns1}^{1*} & \xrightarrow{\sigma_{n1}^1} & \psi_{n1}^1 \end{bmatrix} & \Psi^1 \xrightarrow{\epsilon^1} \rho^1 \\ Q^{2*} \begin{bmatrix} Q_1^{2*} & \xrightarrow{\sigma_1^2} & \psi_1^2 \\ Q_2^{2*} & \xrightarrow{\sigma_2^2} & \psi_2^2 \\ \vdots & \vdots & \vdots \\ Q_{ns2}^{2*} & \xrightarrow{\sigma_{n2}^2} & \psi_{n2}^2 \end{bmatrix} & \Psi^2 \xrightarrow{\epsilon^2} \rho^2 \\ & \vdots \\ Q^{nr*} \begin{bmatrix} Q_1^{nr*} & \xrightarrow{\sigma_1^{nr}} & \psi_1^{nr} \\ Q_2^{nr*} & \xrightarrow{\sigma_2^{nr}} & \psi_2^{nr} \\ \vdots & \vdots & \vdots \\ Q_{nnr}^{nr*} & \xrightarrow{\sigma_{nnr}^{nr}} & \psi_{nnr}^{nr} \end{bmatrix} & \Psi^{nr} \xrightarrow{\epsilon^{nr}} \rho^{nr} \end{bmatrix}
$$

Figure 11.8: Diagram of Sense Perception. The proximal stimulation given by the perceptive environment, PE, is sensed through the quantities of the sensory system. Their values are equalized (equalized values are indicated by asterisks) and considered by the proximal information processes solving the *singularity equivalences*. Once the values of the singularities are solved, the *cognitive equivalence* is solved for each set of singularities, leading to the observation of the referents.

The part of the environment which is represented is that which satisfies two conditions: first, that the environmental correlation can be resolved. Second, that it corresponds to the referents of the perceptive process. A part of the environment which corresponds to referent ρ^i will be called *real referent*, and will be indicated by $\mathcal{P}^i$.

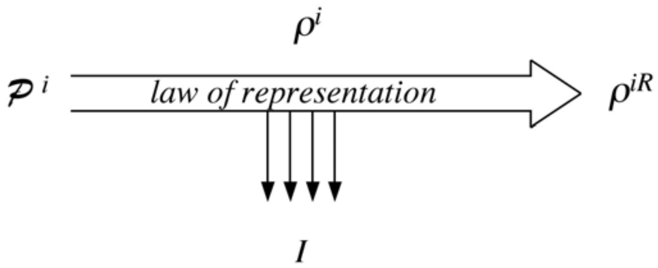


Figure 11.9: Perception of a Referent.

The result of the process of representation is a conceptual entity, which corresponds to $\mathcal{P}^i$ according to the law of representation followed by the process. This entity will be called *represented referent*, and indicated by ρ^{iR}. A diagram representing the process is shown in figure 11.9. We see that a represented referent stands for a particular occurrence of the referent in the real environment. We shall say that the represented referent is an *instantiation* of the referent.

The part of perceptive memory used for a represented referent will be called *perceptive memory unit*. All the units that correspond to a single process of perception will be called *represented percept*. Normally, this term will be shortened to *percept*; nevertheless, it must not be confused with the notion of percept introduced previously, which includes the implicit changes in the system generated by a perceptive process.

In summary, we may say that a certain law of representation, which we may represent by $\mathcal{R}$, establishes a correspondence between the state of a part of the perceptive environment called 'real referent,' $\mathcal{P}^i$ and a part of the perceptive memory which we call 'perceptive unit´. In cognitive terms, the perceptive unit stands for an instantiation of referent ρ^i: ρ^{iR}:

$$\begin{aligned} \mathcal{R} : PE &\longrightarrow PM \\ \mathcal{P}^i &\longmapsto \rho^{iR} \end{aligned}$$

11.4.1 Cognitive Overview of Representation

Let us consider a sense referent, $\mathbf{V} = \{\rho^i, \quad i = 1 \ldots n_r\}$. As we mentioned (section 11.3.2,) the concept of *referent* ultimately stands for quantities, time–invariant relations

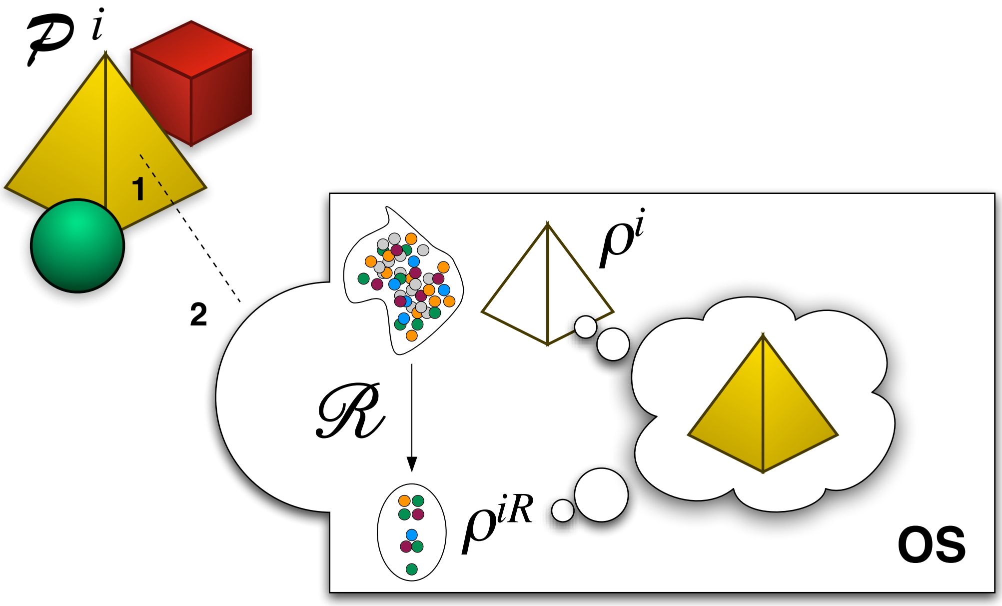


Figure 11.10: Representation of a Referent. The process of representation configures some of the representation resources of the perceptive memory (shown as a disorganized set of circles) according to the law of representation $\mathcal{R}$. The resulting state ρ^{iR} is the represented referent. In cognitive operations of the system, it will be used by the system instead of the real referent $\mathcal{P}^i$. If the law of representation and the representation resources are efficient, ρ^{iR} will simulate $\mathcal{P}^i$ conceptually, situation shown by the coloured pyramid associated to ρ^{iR}. It is an instantiation of ρ^i.

and properties.

For better cognitive analysis, let us consider that referents stand for *objects* and *relations*. Both notions of referent are equivalent. Indeed, an object can be regarded as a group of quantities with a common relation or set of relations. In turn, objects can be related to other objects. Properties are, in fact, objects, understood as ideas.

Therefore, conceiving referents as objects and relations instead of quantities, relations and properties means no loss of generality. It is a more intuitive way for describing referents.

We may consider that a real referent, $\mathcal{P}^i$ stands for a real object and real relations with other objects, which we can express formally as $\mathcal{P}^i = \{O^i, R^i_j, \quad j = 1 \ldots n^i\}$.

In turn, the represented referent, ρ^{iR}, will also be formed by a represented object and a set of represented relations, formally $\rho^{iR} = \{o^i, r^i_j, \quad j = 1 \ldots n^i\}$.

We may see that the law of representation stands for the specification of the correspondences:

$$\begin{array}{rcl} O^i & \longrightarrow & o^i \\ \{R^i_j\} & \longrightarrow & \{r^i_j\} \end{array}$$

11.4.2 Grounded Overview of Representation

Let us explore the correspondence between real referents and perceptive memory units. As we have seen, a perceptive memory unit, ρ^{iR}, stands for the state of a part of perceptive memory. On the other hand, $\mathcal{P}^i$ stands for the state of a part of the environment.

As a system itself, the perceptive memory has an ST–structure and an UC–structure. Let us consider the representation resources used for a perceptive memory unit. We may see that the possible representations that they can hold are the states contained in its state–transition structure.

Let $\{\alpha_k, \quad k = 1 \ldots n_\alpha\}$ stand for the states of the ST–structure, and β^k_l for the possible transitions for each state. The law of representation specifies the correspondence between the states of $\mathcal{P}^i$ and the states of the structure (α_k). It follows that the number of states, n_α, determines the maximum number of real referent states that can be represented. We shall therefore call n_α *cognitive resolution* of the representation.

During the process of representation, the perceptive memory unit is driven by the representation system from an initial state to the final state, ρ^{iR}. We may realize that this may imply following a certain sequence of transitions (β).

We might realize that the ST–structure of the perceptive unit stands for the states and transitions it can hold within the constraints imposed by the UC–structure. The UC–structure determines on one side the possible configurations of the unit (its possible states,) and on the other side the transitions between them. We may realize that, ideally, the more possible transitions from each state, the more direct the process of representation will be.

Part IV

Exemplary Systems

Chapter 12

Driver Attention Monitor

Two of the leading areas of research in automotion are focused on Advanced Driver Attention Systems (ADAS) and in monitoring of passengers and vehicle environment. ADAS identify potentially dangerous situations in driving and warn the driver. Involuntary lane departure warning systems are an example of ADAS. Passenger and environment recognition systems are intended to identify and recognize objects, persons, and situations relative to the vehicle, in order to make it automatically react in the most adequate way to maximize safety.

In this example, we are going to analyze an ADAS for driver attention monitoring (DAM).[1] It is an embedded system, implemented on an Altera Cyclone EP1C3 FPGA. A prototype board was designed for the system (see figure 12.1,) which provided accessible ports for capturing intermediate signals in the laboratory (see figures 12.3 and 12.4.)

The system consists of three parts: a low cost PAL camera installed on the dashboard of the vehicle facing the driver, a board where the FPGA is mounted, and an elementary, acoustic warning system. We are going to abstract the main functional, perceptual aspects of the system, ignoring details as the Analog/Digital Conversion (ADC) and the actual warning to the driver.

From a systemic point of view, DAM can be easily analyzed as a node, in which most of the activity is perceptual. Core processing is limited to inferring the level of the driver's non-attention and the generation of an alarm level. The efferent element is the elementary HMI (Human-Machine Interface) developed for real-testing the application aboard the vehicle.

[1] Félix Moreno et al. *A low–cost real–time FPGA solution for driver drowsiness detection*, [MAHP03]. Instituto Universitario de Investigación del Automóvil [Automobile Research Institute], Universidad Politécnica de Madrid. Figures and explanations are included with kind permission.

Figure 12.1: Prototype board for the DAM.

12.1 Description of the System

In order to describe the operation of the system, an activity diagram is shown in figure 12.2. As mentioned above, the operation is explained following the notion of node introduced in section 7.4.6, p. 128. It can be observed that the system is mainly a perceptual one, with scarce core and efferent processing.

Following the operation of the system, we see that the first step of the process is an A/D conversion, carried out following the ITU-R 601 Recommendation for PAL systems. As a result, the system receives a frame of 720×576 pixels. A new frame is received each 40ms. Colour information is discarded; the system considers luminance only. In order to minimize noise, the image is median-filtered. Edges are then extracted with a Sobel filter, giving way to a black and white image, see figure 12.3.

Not all the frame is analyzed. From the 720×576 pixel input, a region of analysis of 100×204 pixels, including the eyes of the driver, is extracted. The rest of the image is ignored; see figure 12.4. This lightens the computational load, enabling real–time performance.

Extracting a region implies the need for tracking the position of the eyes. The system executes a tracking algorithm for this, working on a three–frame basis. A speed vector is calculated from the first two frames to represent the movement of the eyes, and predict the position in the third frame.

In the normal cycle of operation, the system analyzes whether the driver is either opening or closing the eyes. If the same situation is detected for over U_{natn} frames, the system interprets a loss of attention and warns the driver. An opening eyes situation is detected by comparing the number of white pixels in the region of analysis, n_{wp}, with a threshold value U_{eyes}:

$$n_{wp} \quad \begin{matrix} > \\ < \end{matrix} \quad U_{eyes} \quad \begin{matrix} \text{openingeyes} \\ \text{closingeyes} \end{matrix}$$

There are two events which can drive the system out of the normal cycle of operation. A change in lighting conditions, due to passing through a tunnel for example, makes the system momentariliy stop analyzing the driver. Instead, it measures the new ambient light, and adjusts its parameters to it. This lasts U_{env} frames. The number of white pixels in the region is counted in each of these frames. The new light conditions are characterized by the maximum and minimum numbers registered in the sequence: n_{MAX} and n_{MIN}. These values are used to calculate the threshold value for distinguishing the opening/closing eyes situations:

$$U_{eyes} = \frac{n_{MAX} + n_{MIN}}{2}$$

The other event that drives the system out of the normal cycle of operation is losing the position of the eyes for a certain interval of U_{noeyes} frames. This might occur if the driver blinks for a long time, or moves the head out of the scope of the camera. The system interprets this as a loss of attention and might alert the driver.

12.2 Perceptual Analysis of the System

The DAM system is a representative example of a commercial system at prototyping stage. General concepts of perception such as the ones proposed in part III were not considered for designing it: the final architecture and parameter values resulted from long series of experiments and intermediate developments [MAHP03]. As expected, the concepts of perception and perceptive systems introduced here are implicit in the design: modules and functions have been designed according to technical criteria exclusively, regardless their perceptive function.

We shall assume the sensory system of DAM is a progressive, digital CCD camera, sending a 720×576 pixel frame at PAL rate (40ms.) This will enable us to abstract the ADC conversion for clarity. [2]

12.2.1 Referents and Senses

From the previous description of system operation, we might realize that it operates with seven concepts, as represented in figure 12.5. These concepts are the *referents* of the

[2] Although the real system was implemented with a PAL camera, laboratory versions were made using a SONY XC-EI50 CCD device. This enhanced performance (response time) by eliminating the ADC conversion and by including infrared operation, which enabled night–monitoring. The solution was not adopted, however, because the system was intended to be of low cost, for mass production in vehicles.

system. Of the seven referents, only U_{eyes}, *brightness* and *non–attention* are explicit. The rest are implicit, and appear as separate parameters.

The system carries out multiple perceptive processes; each *sense* has its own. These processes take place at different time–resolution levels. The perceptive processes are described in figure 12.6, listed according to their time-resolution.

12.2.2 Singularitites

The set of *sensory system quantities*, $\mathbf{Q}^{\mathbf{ss}}$, corresponds to the 720×576 pixels of the camera. Luminance extraction, median, and sobel filtering are *equalization* processes carried out by the system.

As it was mentioned in the description of the system, only a 100×204 pixel *region of interest, RoI*, is considered by the system. *RoI* is the subset of sensory system quantities upon which each of the singularities operates, which in our notation we have indicated by Q^i_k, for all referents i, and all referent-singularities, k. We realize that in DAM all sets coincide:

$$Q^i_k = RoI, \quad \forall i, k$$

and, therefore:

$$\bigcup_{k=1\dots n_i} Q^i_k = \quad Q^i \quad = RoI, \quad i = 1\dots 6$$

Referent	**Set**	**Singularities**
ρ^1:eyes_state	$(\dagger)\Psi^1$	ψ^1_1 : no_white_pixels
ρ^2:eyes_position	Ψ^2	ψ^2_1 : h_max_no_white_pixels ψ^2_2 : v_max_no_white_pixels
ρ^3:eyes_estim_pos	$(\dagger)\Psi^3$	ψ^3_1 : h_max_no_white_pixels ψ^3_2 : v_max_no_white_pixels
ρ^4:env_cond	Ψ^4	ψ^4_1 : no_white_pixels ψ^4_2 : max_no_white_pixels ψ^4_2 : min_no_white_pixels
ρ^5:eyes_thresh	Ψ^5	ψ^5_1 : no_white_pixels ψ^5_2 : max_no_white_pixels ψ^5_2 : min_no_white_pixels
ρ^6:brightness	Ψ^6	ψ^1_6 : no_white_pixels
ρ^7:non-attention	$(\dagger)\Psi^7$	ψ^7_1 : no_white_pixels

Table 12.1: Referent Sets of Singularities in DAM.

The correspondence between referents and singularities can be observed in table 12.1, through the *sets of singularities*. It must be remarked that the set of singularities for each referent stands for the quantities upon which it ultimately depends. However, it gives no indication to whether the referent is perceived directly or indirectly through another, mediatory referent. In the particular case of the DAM system, we have indicated by a '†' symbol that referents 1, 3 and 7 depend indirectly from their corresponding referent sets. The actual dependence can be clearly observed in figure 12.6. Referent 3 is calculated from representations of referent 2. Referent 7 is perceived from representations of referent 1.

12.2.3 Equivalence Functions and Perceptual Maps

Singularity equivalence functions establish the correspondence between sensory system quantities and singularities. The names of the system singularities, listed in table 12.1, have been chosen in order to represent their corresponding equivalence functions. In this system they are all functions which either count the number of white pixels, or which calculate minimum and maximum numbers in a sequence. We can see that singularities are, in reality, parameters describing the readings of the sensory system.

Cognitive equivalence functions perceive objects parting from singularities. The seven cognitive equivalence functions in the system are shown in table 12.2.

Dependence	Function
$\epsilon^1(\rho^5, \rho^5)$	$\rho^1 =$ eyes_opening if $\rho^6 > \rho^5$ $\rho^1 =$ eyes_closing if $\rho^6 < \rho^5$
$\epsilon^2(\psi_1^2, \psi_2^2)$	$\rho^2 = (\psi_1^2, \psi_2^2)$
$\epsilon^3(\rho^2(1), \rho^2(2))$	$\rho^3 = \rho^2(2) + (\rho^2(2) - \rho^2(1))$
$\epsilon^4(\psi_2^4, \psi_3^4)$	$\rho^4 = (\psi_3^4, \psi_2^4)$
$\epsilon^5(\psi_2^5, \psi_3^5)$	$\rho^5 = \frac{\psi_3^5 + \psi_2^5}{2}$
$\epsilon^6(\psi_1^6)$	$\rho^6 = \psi_1^6$
$\epsilon^7(\{\rho^1(i), \quad i = 1 \ldots i\})$	$\rho^7 = i$ if $\rho(j) = \rho(1), j = 2 \ldots i$ $\rho^7 = 0$ otherwise

Table 12.2: Cognitive Equivalence Functions in DAM. Numbers in brackets indicate frame number in a sequence of frames.

Relations among referents and singularities can be represented graphically. This representation is a perceptual map. A possible perceptual map for DAM is shon in figure 12.7.

This map shows that referent 1, the state of the eyes, depends on referents 5 and 6, the threshold for distinguishing between opening and closing and the brightness of the environment. In turn, referent 7, the level of non–attention, depends on referent 1: whether the eyes are being opened or closed. The predicted position of the eyes, referent 3, depends on the real position of the eyes.

The asterisks beside the dependences between referents 1 and 6 and singularity $\psi_1^{1,4,5,6}$ indicate that these dependences are mutually exclusive. Both have been shown for didactic purposes, in order to show a case where some singularities depend on others, such as $\psi_2^{4,5}\psi_3^{4,5}$ depending on $\psi_1^{4,5}$. However, a proper perceptual analysis of DAM would have considered the following referents:

- ρ^1: As it is, the state of the eyes.
- ρ^5: As it is, but would become dependent on $\rho 6a$ instead of the current singularities.
- ρ^6: As it is, the current environmental brightness.
- ρ^{6a}: The environmental current range of brightness, with limit values given by the minimum and maximum values of ρ^{6R} in a U_{env} frame sequence.

This analysis and the one explained are compatible for the particular case of the DAM system. In general, however, it will depend on how the system is actually implemented. In DAM, for example, the actual memory register used for ρ_1^1, ρ_1^4, ρ_1^5, and ρ_1^6 is the same. In fact, the number of white pixels in the RoI is calculated only once per frame, and subsequently used by all functions. It can therefore be considered to be a unique singularity, such as it has been done in the figure: $\rho_1^{1,4,5,6}$. The same with the other grouped singularities. This would not necessarily happen with a different implementation.

It follows from the concepts of referents and singularities that a singularity should not depend on a referent, although referents can depend on referents and singularities, as referent 1 in the figure. This is because singularities parameterize the readings of the sensory system.

12.2.4 Elements of the Node

The activity of the elements of the node can be best represented on the activity diagram, as shown in figure 12.8. The shaded parts highlight the activities of the different elements.

Only part of the integrative element can be observed in the figure, because shared memory registers for passing information between elements, memory elements in which the U_{natn} and U_{env} parameters are stored, and cabling between elements are not shown in an activity diagram. In the diagram, the *join* symbols marked 3, 4 and 5 are considered part of the integrative element. The rest are not because they are understood to form part of the corresponding element's structure.

If the DAM system would include devices for optimizing the system behaviour, for example a run–time controller to guarantee real–time, or a fault monitor for implementing fault–tolerance, they would form part of the integrative element.

Within the perceptor (afferent element in the node terminology) the equalization functions are indicated by a bounding box labelled 'eq'; cognitive equivalence functions are indicated by ϵ. The result of all functions is strictly based on the readings of the sensory system, except for ϵ^3, the prediction of the position of the eyes in the next frame. This is indicated with a '†'. It can be observed in figure 12.6, that ρ^3 is inferred, and a deliberative subprocess takes part, marked 'dv.'

The DAM system implements ϵ^7 straightforwardly. It is a counter of the number of consecutive frames with no alteration in the state of the driver's eyes. This is, however, an elementary form of interpreting the driver's degree of non–attention. A more developed function reflecting psychological aspects could be used.[3] This function could eventually be implemented as an inference process by an expert system, pattern matching against a database, decision–tree, etc. Any of these implementations would stand for a deliberative process such as the 'dv' mentioned regarding ϵ^3. This has been indicated with a † symbol.

The actual ϵ^7 analyzed here uses knowledge implicitly. It is the knowledge of the engineer of the system. According to this knowledge, there is a direct correlation between the degree of non–attention and the number of frames of constant state of the eyes. This correlation is linear, and therefore it is implemented as a counter. The core element, in decision labelled '6,' uses explicitly knowledge in the parameter U_{natn}. This knowledge expresses the threshold level of danger, and triggers an alarm. As in the case of ϵ^3 and ϵ^7, a more sophisticated core processing, deliberative process could be developed.

It must be mentioned that perceptual analysis of a system is not unique, and that several, different analyses may describe the same system equally well. In fact, the result of analyzing perception in a system (distinguishing singularities from referents, the perceptor from the other node elements, etc.) comes from observing the system from a given point of view. Some possible points of view are: implementational, architectural, cognitive and functional among others. Each point of view determines which aspects of the system are emphasized: how the system is actually built, relations among its major parts, dependences among the concepts that it uses or what the system parts are supposed to do, for example. It might prove necessary to analyze the system trying to reflect what the designer actually wanted to implement, which might be very different from what actually exists.

The DAM system has been analyzed here from an implementational–cognitive point of view, trying to reflect how it actually works in terms of cognition (it is an example of how perception is implemented.) As it has been mentioned previously, the analysis has also tried to show different cases of relations between singularities and referents in a real system, although this has made it more tedious than it would be necessary for this system.

[3] In fact, it was actually scheduled for future developments of the system.

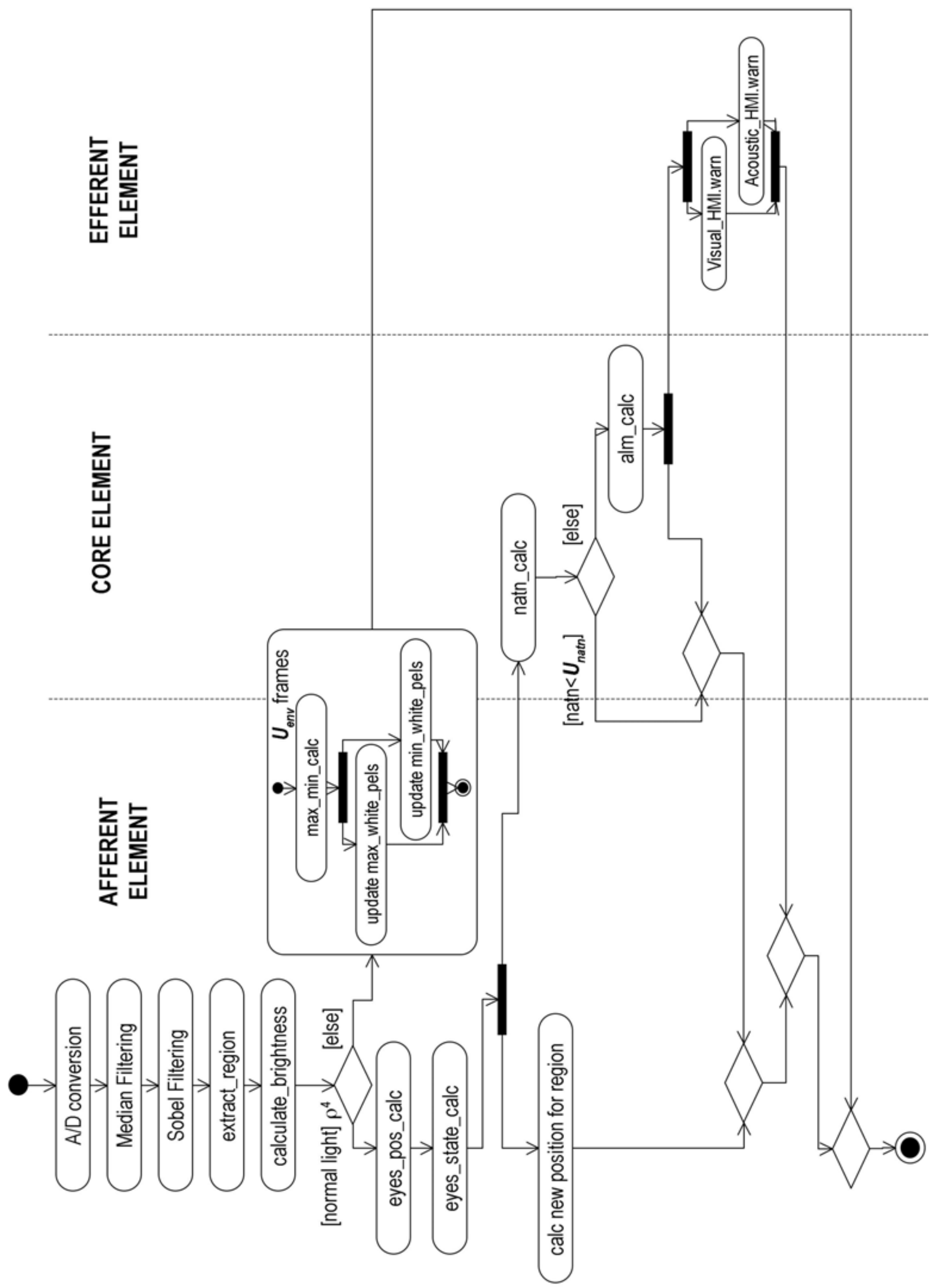


Figure 12.2: Simplified Activity Diagram of the DAM System.

Figure 12.3: Image preprocessing in the DAM system.

Figure 12.4: Region of Analysis in the DAM System. The image is shown by reading intermediate signals directly from the board on a laboratory, as they are not accessible in the real, in–vehicle configuration.

eyes
- opening / closing $\rightarrow \rho^1$
- position / non–exist $\rightarrow \rho^2$
- estimated position $\rightarrow \rho^3$

environmental conditions
- light conditions $\rightarrow \rho^4$
- U_{eyes} $\rightarrow \rho^5$
- brightness $\rightarrow \rho^6$

non–attention
- $value \rightarrow \rho^7$

Figure 12.5: Cognitive Referents of the DAM System. The first three referents with which the system operates characterize the eyes of the driver, the following three, the environmental conditions and the last, the level of non-attention of the driver. The system has a sense for each of the seven, which are treated separately.

Frame Level:

$$\text{frame (1)} \xrightarrow{\sigma_1^{1,6}} \text{no\_white\_pixels} \rightarrow \left[\begin{array}{l} \xrightarrow{\epsilon^{1\dagger}} \text{eyes\_opening/closing} : \rho^{1R} \\ \xrightarrow{\epsilon^{6}} \text{brightness} : \rho^{6R} \end{array} \right.$$

2-Frame Sequence Level:

$$\left. \begin{array}{l} \text{frame (1)} \rightarrow \left[\begin{array}{l} \xrightarrow{\sigma_1^2} \text{h\_max\_no\_white\_pixels} : \psi_1^{2R} \\ \xrightarrow{\sigma_2^2} \text{v\_max\_no\_white\_pixels} : \psi_2^{2R} \end{array} \right] \xrightarrow{\epsilon^2} \text{eyes\_pos(1)} : \rho^{2R} \\ \text{frame (2)} \rightarrow \left[\begin{array}{l} \xrightarrow{\sigma_1^2} \text{h\_max\_no\_white\_pixels} : \psi_1^{2R} \\ \xrightarrow{\sigma_2^2} \text{v\_max\_no\_white\_pixels} : \psi_2^{2R} \end{array} \right] \xrightarrow{\epsilon^2} \text{eyes\_pos(2)} : \rho^{2R} \end{array} \right] \xrightarrow{\text{dv}} \text{eyes\_pos}(\hat{3}) : \rho^{3R}$$

12-Frame Sequence Level:

$$\left. \begin{array}{lll} \text{frame (1)} & \xrightarrow{\sigma_1^{4,5}} & \text{no\_white\_pixels} : \psi_1^{4R,5R} \\ \vdots & \vdots & \vdots \\ \text{frame (12)} & \xrightarrow{\sigma_1^{4,5}} & \text{no\_white\_pixels} : \psi_1^{4R,5R} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{max\_no\_white\_pixels} : \psi_2^{4R,5R} \\ \text{min\_no\_white\_pixels} : \psi_3^{4R,5R} \end{array} \right] \rightarrow \left[\begin{array}{l} \xrightarrow{\epsilon^4} \text{light\_conditions} : \rho^{4R} \\ \xrightarrow{\epsilon^5} U_{eyes} : \rho^{5R} \end{array} \right.$$

U_{natn}-Frame Sequence Level:

$$\left. \begin{array}{lll} \text{frame (1)} & \longrightarrow & \text{eyes\_opening/closing} : \rho^{1R} \\ \vdots & \vdots & \vdots \\ \text{frame } (U_{natn}) & \rightarrow & \text{eyes\_opening/closing} : \rho^{1R} \end{array} \right] \longrightarrow \text{non–attention} : \rho^{7R}$$

Figure 12.6: Perceptive Processes in the DAM System. The ‘†’ symbol in a cognitive equivalence function indicates indirect dependence of the shown inputs. Some singularity equivalence functions which coincide for different referents have been grouped, such as 1 and 6, and 4 and 5. ‘dv’ stands for ‘deliberative,’ core processing.

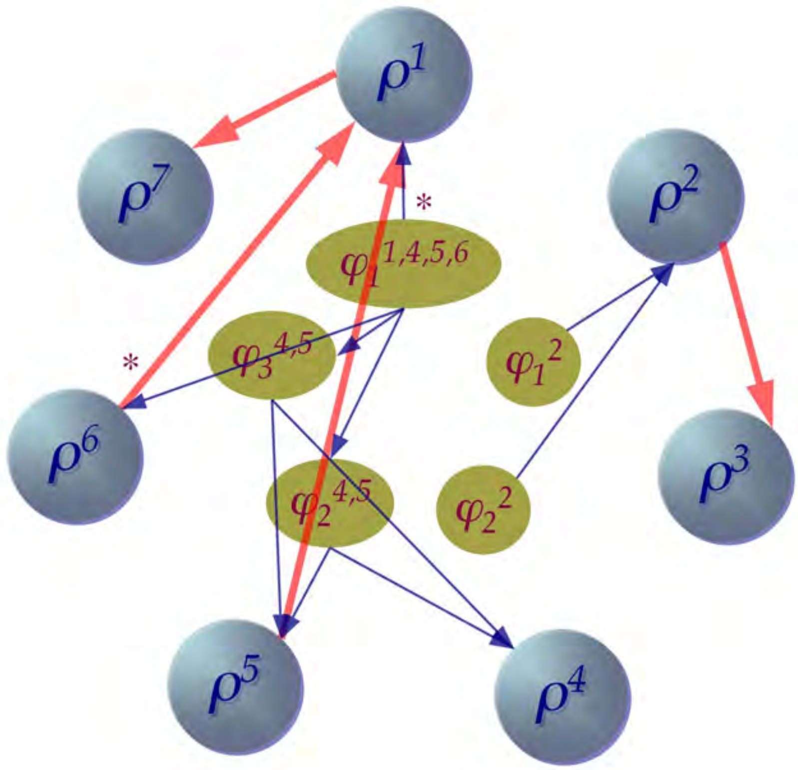


Figure 12.7: Perceptual Map of the DAM System. Dependences between singularties and between singularities and referents are shown by thin arrows. Dependences between referents are shown with thick arrows.

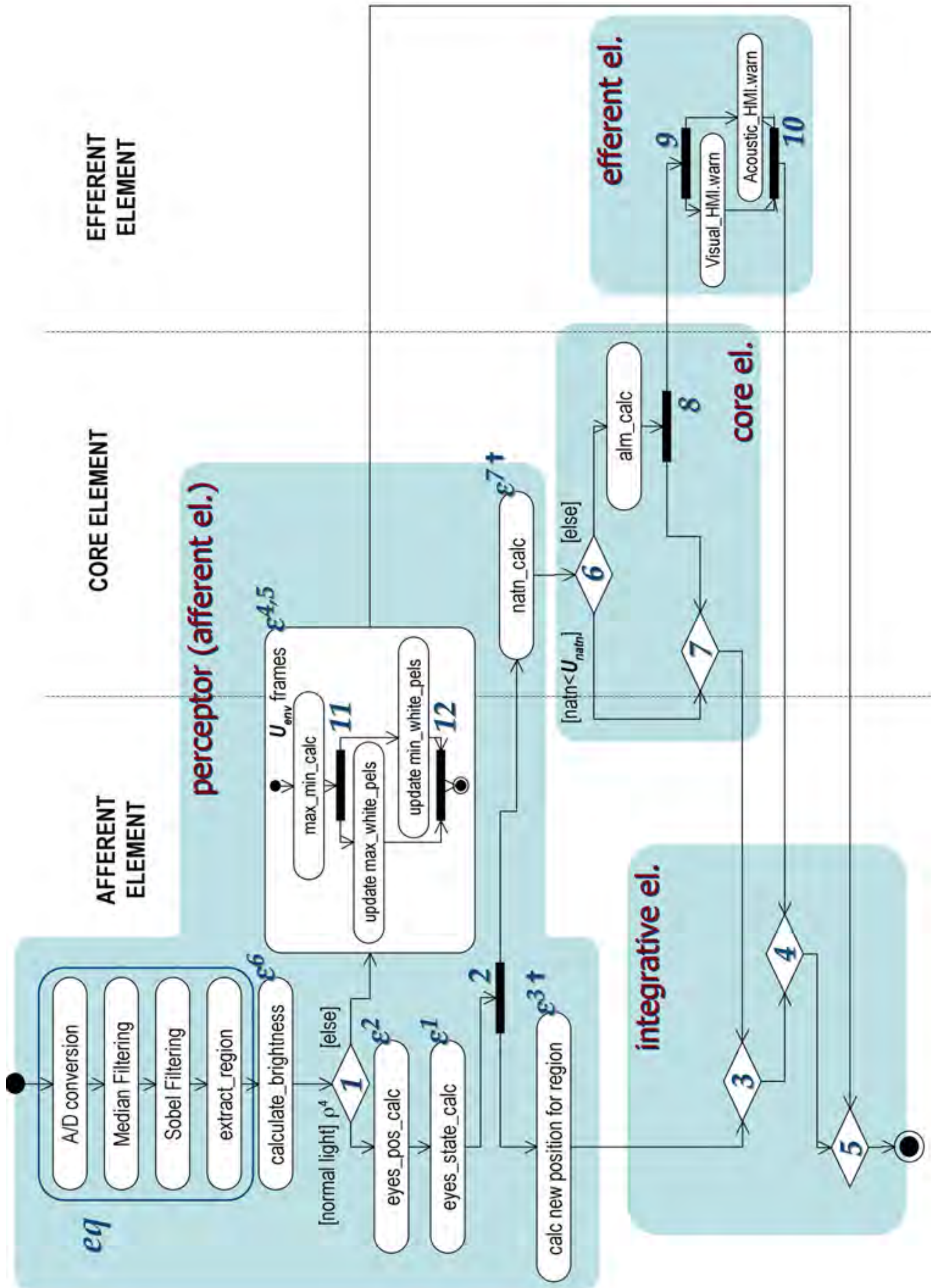


Figure 12.8: Element Activity Diagram of the DAM System. Shaded regions outline distinguish the different element activities: afferent, core, efferent and integrative.

Chapter 13

Complex Intelligent Process Control

this chapter was contributed by Ricardo Sanz

We can use the abstract model proposed in this thesis to recapitulate past works and to extract designs and experiences that can be put under the light of the ASys general perspective. This is the case of the CONEX system [SJG$^+$92], a complex intelligent controller for the cement industry.

13.1 The Context of Intelligent Process Control

While the development of control technology for industry is continuous, the use of truly advanced process control is still quite uncommon in complex process plants. This fact is sometimes justified by the well known gap between research and production people; but, from the point of view of engineering, plant personnel is well aware of the many possibilities of advanced control and the causes of the reduced use may be found elsewhere: in the lack of confidence in the returns they would obtain from using such technologies.

This lack of confidence is due sometimes to system brittleness and sometimes to the operational requirements derived from such advanced technologies. Complex process control systems need specialized personnel for their operation, very specialized personnel for their installation and very very specialized personnel for understanding their operation (specially when it is far from nominal conditions). The effective enhancements that any increment in control technology can produce in plant performance are usually not big, and advanced control system are usually unable to cope with most of the problems that more conventional control systems cannot deal with (abnormal plant behaviours in general).

These problems appear when plant status and dynamics violate any of the assumptions made when designing the control system. These assumptions have two clear origins:

- Incompleteness of the systems analysys: Assumptions that are made inadvertently.
- Limitations of the control technology: Assumptions necessarily made due to inabilities of the control system technology.

The final implication of these design assumptions is that the on-line control system does not react adequately to the situations bypassed in the analysis phase of the control system development. To cope with the always existing possibility of experiencing such situations, the engineering solution is to use a *supervised control structure*. The final responsibility of dealing with those abnormal situations is of the intelligent *operator* and *process engineer*; the humans that monitorise the operation of the plant/control system.

A main objective of using artificial intelligence techniques in these systems is having most of those situations under computer control by mapping humanly characteristics into the control system. The advantages of computer monitoring and control over human ones are clear, and artificial intelligence can be a way to translate human skills to the computer. The need for human supervision does not completely disappear, but the responsibility of dealing with abnormal but frequent situations is of the intelligent control system.

The research directions in intelligent process control have been quite heterogeneous, but the main ones are the following:

- Computerization of human skills: the classical example is the *expert control system*.
- Enhancement of control system performance using AI technology: e.g. expert PID tuning.
- Extending situation coverage: dealing with faults as does an on-line diagnosis system.
- Search of reactiveness and predictability: by using special architectures and techniques.
- Introducing new capabilities in the control system: e.g. advanced pattern recognition.
- Improving system robustness: e.g. using fuzzy sensor fusion..

13.2 Integrated Complex Intelligent Controllers

In general, it is not possible to reach all these objectives using only one technique — e.g. only expert systems— but using a set of them integrated in one single application. This is the case of the CONEX system, where several AI technologies are integrated to achieve improved control [San90]. The need of integrated intelligent control systems has redirected some research effort to the elaboration of mechanisms and architectures that enable this unified operation of the subsystems.

For a better understanding of this type of systems and technologies we will refer to three kinds of things: basic intelligent control technologies, real-time integration architectures and integration methodologies.

Basic technologies are the working elements of integrated applications. They perform the final tasks that make the control system useful. Integration architectures are global designs of applications composed by a set of working elements. The architecture specifies the roles of subsystems and the interactions between them. Finally, integration methodologies specify how to use software technologies that enable data and services interchange between subsystems.

13.2.1 Basic Intelligent Control Technologies

Technologies coming from all the areas of artificial intelligence and conventional software had come to try their capabilities against plant control problems. Some of the most promising ones are expert systems, fuzzy systems and neural networks.

Expert system technology once was the flagship of intelligent control. This is due to the possibilities of expert system shells to model human knowledge, which has made possible quick implementations of computerized expertise-based control. The R+D effort in this area has been quite big and as a result there are several commercial expert system shells that can be employed with success in real-time applications. Now expert system technology based on elaborations of classical production systems are common technological platforms for strategic and tactical control systems.

Fuzzy systems offer a way to enhance the behavior of somewhat elementary controllers by introducing imprecise knowledge in them. This kind of knowledge embedding was difficult with classic controllers, because these controllers are crisp but the human brains and the reality they perceive are usually are not. Control knowledge affected by uncertainty can be managed and used in real-time [MACJ$^+$95]. We all have seen examples of this technology even in the daily press.

On the other side, neural networks offer a way to get some form of computational knowledge from raw data. Using this technology it is possible to develop advanced systems (identification, control, prediction, etc.) when there is no explicit knowledge available but tons of data. This technology also addresses the time varying character of systems under control, being able to autotune itself to new system characteristics.

13.2.2 Real-time Intelligent Control Architectures

The architectural design of an integrated intelligent control system is the main activity in the construction of integrated intelligent controllers because of its influence in development and final performance [FW95]. If the architecture does not provide the support for a particular kind of behavior, the final system can hardly have it. The architecture specifies the subsystems that will compose the final system, their role and the interfaces between them. The structures most commonly employed for IICSs are agent-based architectures and blackboard architectures. In [SGJ+94] a view of the evolution of real-time intelligent control architectures can be seen.

Non-trivial architectures imply cooperation between subsystems. These subsystems are usually called agents in agent based systems and knowledge sources in blackboard systems. Using the ASys terminology we would say that they are *nodes*. Node cooperation is achieved by means of data interchange or service offering-request to build up *integrated system directiveness*. A big part of the design effort is put on the specification of how this cooperation will be achieved and how from this reverse analysis subsystem directiveness is derived. The use of complex architectures should be considered carefully, due to the hardness of making predictable applications using them.

Blackboard architectures are architectures in which the global control of the application has some feeling of opportunistic reasoning and subsystems access a central repository of information that contains the facts related with the problem in hand shared by those subsystems. Blackboard architectures have a star topology, with a special control role performed by the blackboard manager. These blackboard systems —in a simple analysis— may seem to violate basic ASys structuring considerations concerning the organization of the perceptual systems (*i.e.* the separation of blackboards from knowledge sources). However, this separation is just an exemplification of the varying forms that cognitive system nodes may have and that in many cases render degenerate nodes to maximise the capability of achieving a particular supergoal of the system while keeping the subgoal structures of elementary nodes.

Some intelligent control systems a blackboard architecture enhanced with real-time extensions to improve the predictability an real-time behavior of blackboard based applications. This is a clear example of a *perceptive system* as was described in Chapter 11.

On the other side, agent based architectures —like CONEX— are characterized by the division of responsibilities and partial independence of subsystems. There is no central repository of information, and couplings between subsystems are less tight than between subsystems and the blackboard in blackboard based systems. The paradigm of agent interaction is typically a client-server policy. As an example, Fedderwitz and Wittig [FW95] propose a mechanism for further goal decomposition in system nodes —agents— so activities can still maintain the real-time properties needed of expert systems integrated in control systems.

13.2.3 Integration Methodologies

Integration methodologies provide the software support for interaction between subsystems. Integration methodologies are somewhat related with architectures, because they offer ways of interaction that are natural to specific architectures. But it is possible to think in an architecture based in whatever integration methodology we want.

An integration methodology can be viewed at several levels of resolution. The two most important ones are the conceptual level and the implementation level. Conceptual level specifications are related with the kind of things that subsystems interchange. Implementation level specifications are related with the actual software implementation of the interchanges.

There are several integration models in this area, but there is no established standard at the conceptual level, and each integrated application uses the model that is more suitable to its particular needs. Most of the time this is an application specific specification. The paper of Alarcon et al. [ARMA+94] proposes such a kind of conceptual specification for real-time, blackboard based applications from the cognitive standpoint.

On the other side, implementation level specifications are more standardized, because its technologies are employed in most applications. The technologies employed are based on language resources, operating system resources and network resources. Examples of this kind of technologies are: memory sharing, interprocess communication and distributed services.

13.3 The CONEX System

CONEX is an architecture for vertically-integrated, plant-wide, intelligent control. The CONEX system interacts with the plant (typically through a DCS[1] in real settings) and with the human personnel in charge of plant operation as well as with other external systems that may provide some functionality to the control system (*e.g.* an automated laboratory).

CONEX stands for EXpert CONtroller (*CONtrolador EXperto* in Spanish). The CONEX architecture was developed by Sanz [San90] and was initially used to improve kiln operation in cement production plants[2]. System architecture, however, was defined for a wider domain of control systems, *i.e.* large-scale, continuous process plants and was later used in research implementations of process controllers.

13.3.1 The Rotary Cement Kiln

Cement kilns are used for manufacturing Portland and other types of hydraulic cement [Per86]. The cement kiln is the heart of the cement production process and the most critical subsystem in this process due to the critical operational conditions of the pyroprocessing sintering stage.

Cement, the basic ingredient of concrete, is a controlled chemical combination of calcium, silicon, aluminum, iron and small amounts of other ingredients to which gypsum is added in the final grinding process to regulate the setting time of the concrete. Lime and silica make up about 85% of the mass. Common among the materials used in its manufacture are limestone, shells, and chalk or marl combined with shale, clay, slate or blast furnace slag, silica sand, and iron ore. All these materials are ground into powder before entering the kiln where the chemical reaction happens.

A typical process of manufacture consists of three stages:

1. Grinding a mixture of limestone and clay or shale —with some additives— to make a fine "rawmix".

[1] Distributed Control System.

[2] This was done in a R+D project with *ASLAND Tecnología S.A.* that was partially funded by the *Centro para el Desarrollo Tecnológico e Industrial (CDTI)*.

Figure 13.1: Two rotary cement kilns. We can see the cooking tubes —the kiln itself— and the preheating towers. The cooler can be slightly seen at the left.

2. Heating the rawmix powder to sintering temperature in a cement kiln.

3. Grinding the resulting material —clinker— to make the cement that is packed and/or stored.

The rotary kiln (see Figure 13.3.1) consists of a steel tube lined with firebrick. The 1-5m diameter tube slopes slightly (1-4°) and slowly rotates on its axis at between 30 and 250 revolutions per hour. Rawmix is fed in at the upper end, and the rotation of the kiln causes it to gradually move down the tube to the other end of the kiln where the fuel –gas, oil, or pulverized solid — is blown in through the "burner pipe", producing a large concentric flame in the lower part of the kiln tube. As material moves, it reaches its peak sintering temperature experiencing the chemical reactions that render the final products, before dropping out of the kiln tube into the cooler (the material going out of the tube is called *clinker*).

Air is drawn first through the cooler and then through the kiln for combustion of the fuel and then into the heat interchangers of the pre-heaters. In the cooler the air is heated by the cooling clinker, so that it may be 400-800°C before it enters the kiln, thus causing intense and rapid combustion of the fuel.

There are plenty of problems in the control of cement kilns; from the lack of exact theoretical knowledge about the sintering process itself, to difficulties in having good sensor measures in such extreme conditions, to the continuous drift and noise in every process magnitude. Cement kilns were operated by humans —despite the many efforts into building automatic controllers— until AI technology was used. Fuzzy, rule-based technologies are the common technology used today in the implementation of kiln control systems.

13.3.2 The CONEX Context

Figure 13.2 depicts a context diagram for a CONEX system. The CONEX controller interacts with the plant DCS and other agents: the operator, the process engineer, the control engineer, the substratal HW/SW platform and with external technical systems (automated laboratory databases in the implementation case described in [San90]). It is interesting to consider the role that the three humans play in relation with the operation of the system: the operator only has knowledge enough to keep the interaction plant-controller ongoing. If anything abnormal happens, deeper knowledge is required and engineers come into scene; process engineers if the origin of the problem seems to be in the plant and control engineers if the origin of the problem seems to be in the controller.

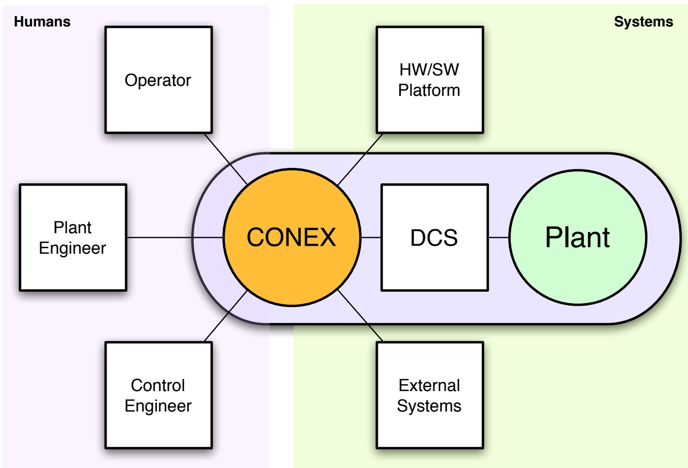


Figure 13.2: System and environment in CONEX.

13.3.3 Overview of the CONEX Architecture

The CONEX architecture as developed in [San90] is composed by nine *nodes* that are organised using a LAYERS design pattern [SSdA+99]. The complex controller for the cement kiln is organised —from an abstract point of view— in five control levels (see Figure 13.2) where the different nodes are playing specific roles (see Figures 13.3 and 13.4).

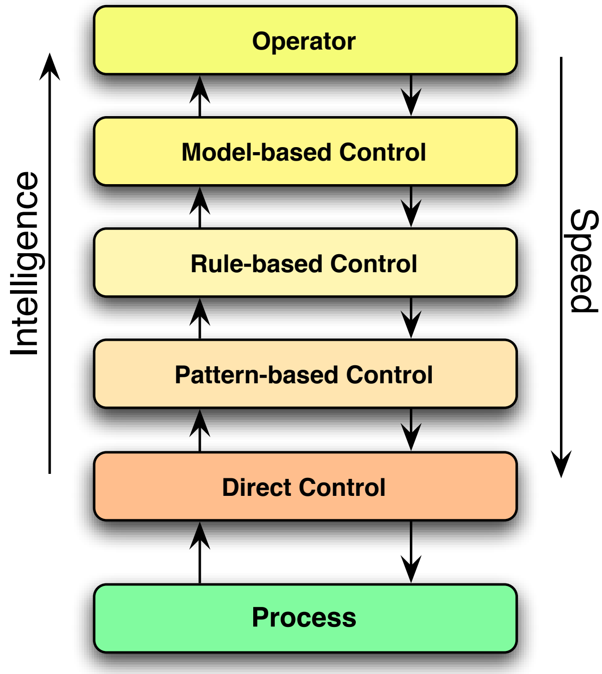


Figure 13.3: Conceptual Control Layers of the CONEX Model.

13.3.4 Cognitive Nodes in CONEX

The CONEX system is composed by nine classes of nodes that pursue specific goals of the global goal structure. The nodes organise in a hierarchical, layered control system (see Figure 13.4).

All CONEX nodes share a common structure that can be analysed in terms of node structure as described in chapter 7. All nodes have two separate *afferent* and *efferent* elements that are based on message-passing systems. The core components of the nodes are what differentiate the nine classes mentioned before. The CONEX node classes are the following:

Process Interface (PI): Interacts with the plant DCS to get real-time information about the plant and the DCS itself, and also serves as a channel for actuation (indeed the CONEX system does not control the plant but the system plant-DCS).

Direct Control (DC): Implements conventional and fuzzy control strategies for simple loop management.

Process Monitor (PM): Implements a pattern-based perceptor and and associated controller.

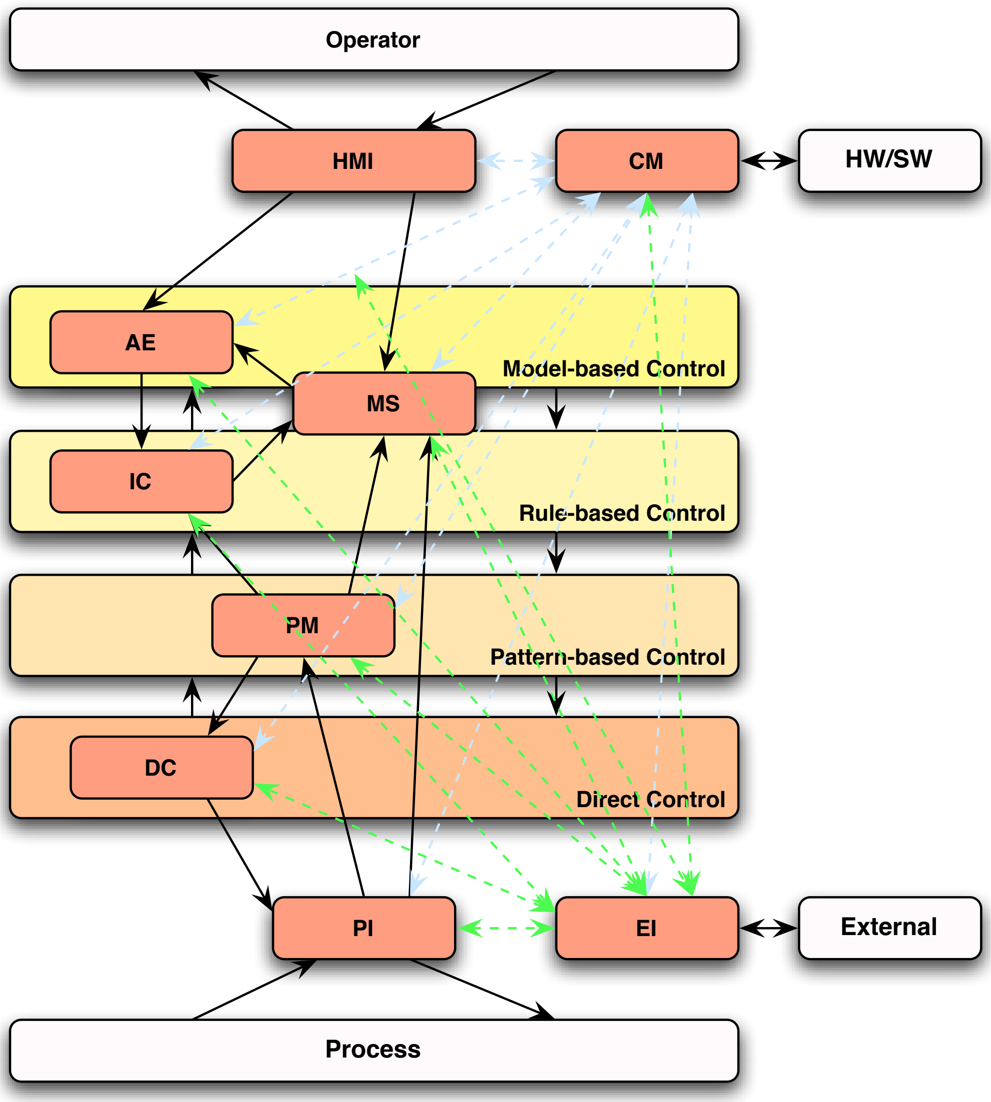


Figure 13.4: Details of the CONEX nodes placed over the conceptual control layers described before. The nodes are: Process Interface (PI), Direct Control (DC), Process Monitor (PM), Intelligent Control (IC), Model and Simulator (MS), Action Evaluator (AE), Human-machine Interface (HMI), External Interface (EI) and CONEX Monitor (CM).

Intelligent Control (IC): Implements a rule-based controller (with multilevel, chained inference processes much more complex that the simple triggering of the pattern-based controller.

Model and Simulator (MS): Stores the multiresolutional representations of the plant state that are used by the different control systems. It uses the state information and deep multiresolutional plant models to make plant behavior predictions.

Action Evaluator (AE): Uses model-based reasoning techniques to evaluate potential actions coming from the rule-based controller, the human operator of the action evauator itself that acts as a model-based controller.

Human-machine Interface (HMI): The interface for the operator and plant engineers.

External Interface (EI): An interface used to access external systems and to open CONEX to interoperation from other agents.

CONEX Monitor (CM): Node that self-monitorises the whole CONEX system.

The node *integrative elements* are common to all nodes and are composed by two subsystems: the message manager and the node status manager. These two subsystems are what make possible the integration of all CONEX nodes into a cohesive architecture.

13.3.5 Perception and Action in CONEX

The perceptual flow in the system uses sensory flows from the outside and also from the inside generating progressively abstract representations. In CONEX, the perceptual processes are organised hierarchically, with each level following the basic structure for a *perceptor* depicted in Figure 11.2. Basically, the overall perceptual process is organised in three levels that render multiple representations of universe state in three classes of *perceptive memories*. They are called N, Q and K representations in the CONEX terminology (N, Q and K stand for numeric, qualitative and knowledge, respectively). Obviously the nature of the *perceptive referents* of all these perceptual processes are multiple (even inside a certain level).

The CONEX multiple perception flows use laws of representation that are adequate for the control technologies integrated in the CONEX system. The CONEX hierarchical representation model describes three major classes of representations and the transformations that they suffer (see Figure 13.3.5).

The action flow also has an hierarchical structure that reflects the *goal structure* of the whole system. This goal hierarchy spans from low level setpoint control to top level strategic objectives like plant stability or maintainability.

13.4 Technologies beyond CONEX

The technologies mentioned so far, have been used with success in several applications. But there exists a vast field of opportunities to new technologies that are emerging in the area of intelligent control. Some of them are strongly related with control, but others are more general.

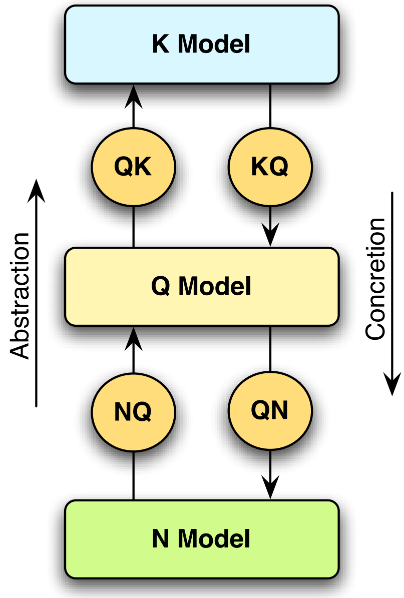


Figure 13.5: Details of the CONEX representational transformations (*laws of representation*) between the conceptual reresentation layers N,Q and K.

Model based reasoning systems are an artificial intelligence technology that focuses in the use of deep knowledge about the systems they are working with. This contrast with the shallow knowledge approach of other technologies such as expert systems or fuzzy control. Deep knowledge is knowledge about the structure and physical relations in the system that is being controlled. The main application of model based reasoning is in the area of diagnosis, however other applications are appearing such as control or data validation.

This kind of deep knowledge can be found in FORMENTOR systems (See paper of Nordvik and Wilikens), an is used to asses operators in unusual and potentially hazardous situations.

Learning is an established technology of AI, but its applicability to real time integrated systems is limited by its reliability. The main problem is robustness of learning. Some learning schemata are more applicable than others. Some examples of application are the use of explanation based learning to learn human behaviors in managing complex plants or the application of genetic algorithms to optimization.

Integration means cooperation, but from some time to now, integration is coming to mean also hybridation. There is a progressive effort in putting together technologies to obtain a product of capabilities. Similar to cooperation, hybridation offers the possibility of obtaining synergistic effects between technologies. The main difference is that in cooperation we have two subsystems that interchange data and services, and in hybridation we have only one system that shares characteristics of both technologies. Examples of hybrid technologies are fuzzy neural networks, geno-fuzzy algorithms, etc (By now, most hybrid technologies seem to be fuzzy-something). It should be clear that the possibilities of hybridation are strongly related wit orthogonality of technologies.

As mentioned before, the integration technology is viewed at two levels: conceptual and implementation. At the conceptual level there are some types of specifications that are emerging to cope with the problem of knowledge sharing and interchange. At the implementation level it should be noted the strengthening of the use of OMA standards. CORBA seems to be the way that near future applications will interact.

From the architectural point of view there is a trend to hybrids between agents and blackboard architectures. The new systems will have distributed transactional blackboards and there will coexist classical passive knowledge sources with client-server based agents.

13.5 Autonomy Principles in CONEX

We can analise the extent to which the CONEX system adheres to the *principles of autonomy*, see section 7.5.3, p.146:

Minimal Structure: CONEX exploits the Knowledge Engine control design pattern in many places to separate generic logic from application knowledge. This reduces the ammount of needed *structure* because most of the control system is *program* in the form of executable knowledge.

Encapsulation: All system logic is encapsulated in the form of modules (high-level objects) that exclusively interact by means of message passing.

Homogeneity: All CONEX modules are based on a single integration platform and share a common base internal architecture. This provides a great degree of homogeneity that is exploited by the CONEX Monitor to self-manage the system state. Also, the common implementation of all node integrative elements give a increased homogeneity.

Isotropy of Knowledge: CONEX tries to eliminate biasing as much as possible by means of plant knowledge integration provided by the MS module. This module contains the perceptual inputs in multiresolutional represeatations that are *shared* by all CONEX *modules*.

Scalability: All modules can be freely deployed across a set of distributed computers. The CONEX Communication Layer middleware provides the means for scaling computing power up to the need of a concrete application. Centralized elements are only used during bootstrap and hence bottlenecks are minimized to communication channels.

Chapter 14

FTMPS: Fault-Tolerant Massively Parallel Systems

This chapter will analyze a real system for *directiveness*, *functional structure*, *nodes*, *cognitive system*, *grounded system* and *cognitive model of the system* among other concepts introduced in chapter 7. Generally, these concepts may apply only in very complex artificial systems. Fault–tolerant, massively parallel systems have been considered for this example, both from a general point of view, mainly structured through [Jal94] and [Cri93], and from the specific point of view of the FTMPS Project [BAB+95], [DVC+94], [VDL+94].

14.1 Introduction: Massively Parallel Systems, Fault Tolerance

The term massively parallel systems, MPS, is used for computer systems formed by multiple hard disks and a number of processors which is typically in the range of thousands. They are used for applications which require intensive number processing such as simulation of natural phenomena and complex system modelling. In spite of the computational power of these machines, processing may take long time intervals; hours, days or longer.

The probability of faults in such systems and operating conditions is not negligible due to the large number of elements, fact which has fostered the development of fault-tolerance mechanisms for them. Fault–tolerant systems continue operation in spite of the failure of one or more of their components. The fault–tolerance mechanisms mask component failures to avoid system-level failures. Ideally, in the event of a fault, the system will exhibit unaltered behaviour with respect to normal operation. This is not always fully possible, and the performance of the system may eventually decrease, but continue. This is called graceful degradation.

The basic terminology regarding system failures is introduced in the following excerpt from [Jal94, p.6]:

> A failure of the system occurs when the behaviour of the system first deviates from its specification. An error is that part of the system which is liable to lead to subsequent failure. If there is an error in the system state, then there exists a sequence of actions which can be executed by the system and which will lead to system failure, unless some corrective measures are employed. The cause of an error is a fault.

There does not exist a unique mechanism or technique for maksing all the types of failures that can occur in a system. Many of the techniques employed involve replicating hardware or software components, introducing significant overheads in inter–component communications, or on–line self–diagnosing. In any case, fault–tolerance may become costly in resources or computational load. Therefore, the degree of fault–tolerance is adjusted to the requirements of each specific system and application. However, fault–tolerance mechanisms usually follow a well–established sequence of phases, regardless the mechanism employed [Jal94, p.9]:

1. Error detection: The presence of a fault is deduced by detecting an error in the state of some subsystem. Information about the failure in the form of error detection code, EDC, is subsequently propagated [Cri93] to the appropriate components in order to be masked.

2. Damage confinement: The damage caused by a detected error has to be confined and delimited. Self–diagnosis mechanisms establish the scope of the damage.

3. Error recovery: The error in the state is corrected. Dedicated circuitry at the hardware level of the system may mask some types of failures, typically at bit level. Mechanisms at operating system level may mask some types of hardware component and operating system failures. At the application level, failures at all other levels can be masked. [Cri93]

4. Fault treatment and continued service: The faulty system components are ceased to be used or used in a different manner so that the fault does not cause failures again.

The example FTMPS system analyzed in the following sections offers examples of error detection, damage confinement, error recovery and fault treatment at different levels of abstraction, from the hardware to the application levels.

14.2 Overview of FTMPS Architecture

The FTMPS project [BAB$^+$95] proposes the structure for a fault-tolerant massively parallel system, FTMPS, shown in figure 14.1. The hardware layer is abstracted from the software layer by a hardware–independent layer, HIL. This layer consists of platform–dependent software, which presents a common interface to the upper, software layer. This upper layer contains control software for run–time process and system management, and the user application software. It is structured following the unifying system model, USM [VDL+94].

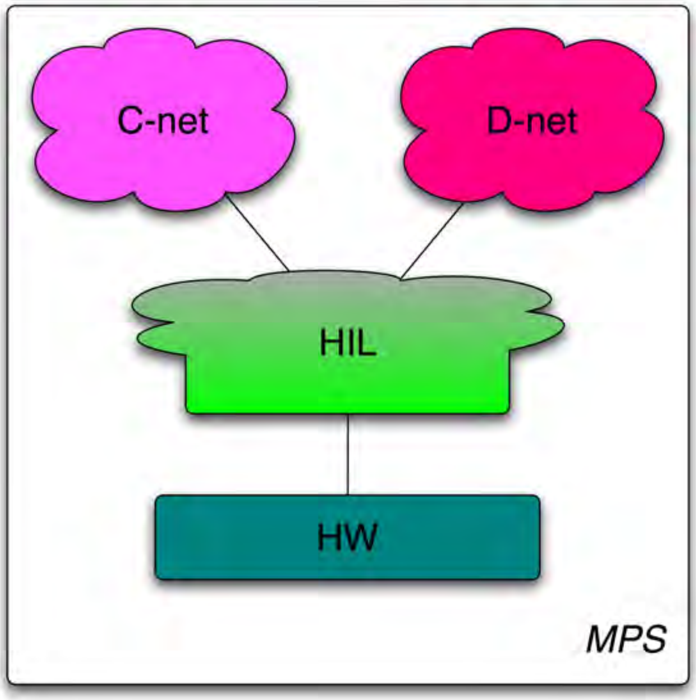


Figure 14.1: Conceptual Layers of the FTMPS Model. Hardware–specific parts are represented with straight–line boxes. Virtual with rounded shapes. C-net and D-net stand for control-net and data-net respectively.

The USM is a logical software model, meaning that the parts and structures it defines may be mapped to a different topology in the hardware layer.

The FTMPS conceptual model has important structural implications:

- The separation of the c-net and d-net layer from the hardware layer increases system *encapsulation*.
- By adding the HIL layer, the dependence between hardware and software is reduced.
- At the same time, the HIL is an increment to the *UC-structre*.

Some of these effects are opposite. The result is that the real structure of the system is reduced as a consequence of separating hardware and software. The *hypothetical structure* is increased by adding HIL. Depending on the specific implementation of the HIL, the *program* would be either reduced or unaffected. In summary, the principle of *minimal structure* has been followed.

As a result of this structure —in comparison with the direct interaction between software and hardware in absence of HIL— it is expected that system *performance* will either remain equal or be slightly reduced, and *adaptivity* significantly increased. We might observe that this is in line with the objective of a fault-tolerant system.

As we have just mentioned, introducing HIL has the advantages of increasing *encapsulation*, reducing *real structure* and increasing *hypothetical structure*. Apart from increasing system *adaptivity*, this last point allows achieving a hardware–independent software layer, which is the effect that was actually desired by the designers:

> The main disadvantage [. . .] is that a system containing a HIL is likely to be less efficient than one which is its functional/hardware equivalent but where the software has been developed with detailed knowledge of the target hardware. However, the potential time, the effort and cost benefits provided by adopting this approach are considerable.
>
> Some serious thought and a little investment up-front are required to get a good HIL specification which is stable across many types of processor and which will incur acceptable runtime inefficiencies only [BAB+95, p.2].

14.2.1 Software Architecture: USM

Specifically, the USM structures software in two categories:

D-net: Stands for Data-net. It includes the applications launched by the user. A specific application is executed by a partition. A partition is a group of nodes dedicated to a single application. A node, in turn, is a group of application processors (also called data-processors), of which some are active and some are spare processors.

C-net: Stands for Control-net. It is the system control software. It provides fault–tolerance mechanisms. There exists a host for the whole MPS, which provides shared services for all the components of the system: global diagnosis, recovery control, and acts as interface with the user. The host is also called global C-net.

Apart from the global c-net, the c-net is formed by a collection of control processors, also called local c-net. A local c-net controls a number of data processors.

As it has been mentioned above, partitions, nodes, data processors, and control processors are different logically, although they can eventually map to the same hardware processors.

The separation between c-net and d-net allows categorizing system objectives in two kinds: *homeostatic* and *application*. The first stand for maintaining system operation within established structural parameters. The second stand for satisfying the specifications for the user applications the system is executing. C-net consists of resources and processes dedicated to homeostatic objectives, and D-net consists of the resources and processes dedicated to application objectives.

The system will strive for maintaining homeostatic behaviour independent from application behaviour, so that the application specifications are satisfied independently from the actual homeostatic dynamics. In case of conflict, preference is given to homeostatic behaviour at the cost of reducing performance.

During normal operation, the *functional structure* of the system is formed by functions derived from the two types of objectives. The correspondence between the functions and the actual resources devoted to them is given by *maps*. Maps are in the form of routing tables that specify the communication topology between hardware nodes, and partition maps that specify the mapping between partitions, control processors and application processors and real harware resources. The FTMPS system has routing and mapping algorithms to adapt the functional structure to system evolution.

In the terminology introduced in part II, routing and planning algorithms are *grounding* mechanisms. They establish the correspondence between the *cognitive system* and the *grounded system*.

In FTMPS, the *cognitive* and *grounded* systems are explicitly separated through the HIL. This layer is a collection of *software drivers* that abstracts the implementation —grounding— details from the cognitive system. It provides efferent channels to application and homeostatic functions. From the point of view of perception, it provides *singularities* and *referents* to the cognitive system.

14.2.2 Hardware Layer

In principle, the FTMPS architecture is platform–independent, provided the HIL is conveniently developed for implementing the USM. However, the sources explicitly mention two platforms upon which experiments have been carried out: Parsytec PowerXplorer, Parsytec GCel, and Parsytec GC/PowerPlus. It is also mentioned that these computers incorporate all features of MPS: scalability, regular system structure and a large number of processing nodes. Details of the different models are out of the scope of this text. A brief description of the PowerXplorer will be given follwoing in order to illustrate the type of hardware architectures treated.

The basic units are clusters of 4 processing nodes, each including a processor, a hardware link and a memory module. Each link allows four connections, which are used to form a grid topology as shown in figure 14.2. The system can be divided into partitions, where one processing node has to be connected to a host, responsible for file access, remote procedure calls, etc.

The exemplary Parsytec PowerXplorer system described shows the main characteristics of the hardware employed for MPS: regularity, scalability and modularity. In

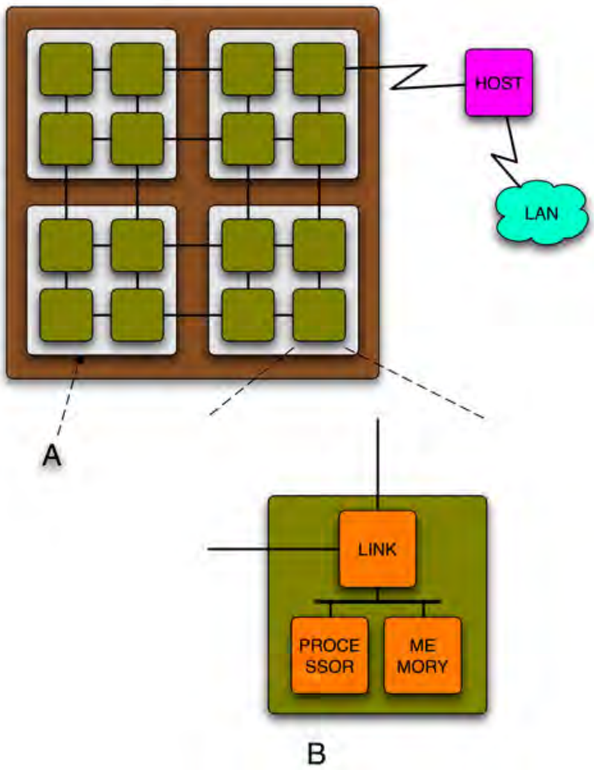


Figure 14.2: Structure of the Parsytec PowerXplorer (after [BAB+95]). Here, a system of 16 processing nodes (B) is shown. Nodes are grouped in clusters of 2×2 (A). Each node (B) has a hardware link of four channels, a processor and a memory module. The resulting structure is a two–dimensional array of nodes. Other models implement three-dimensional arrays.

terms of the *principles of autonomy*, these characteristics stand for *encapsulation* and *homogeneity*.

Encapsulation derives from the fact that the hardware structure is based on nodes, each of which includes processor, memory and communication hardware. A node is therefore self–contained to a high degree. Dependence with other nodes occurs through the communication channels, which form a bounded, well–established interface with the rest of the system.[1]

Homogeneity derives from the identical structure of all the nodes in the system, and the four–channel hardware links included in the nodes. These factors allow building grids of nodes of any size (scalability,) interchanging a node for another and freely re-

[1]It is understood that dependence also occurs in other implicit ways, related to the *real structure* of the platform such as ambient temperature or power supply for example. However, these are omitted for clarity.

assigning the tasks assigned to each node. This versatility is exploited by the *grounding* mechanisms of the system.

14.3 Error Detection

The FTMPS considers six mechanisms of error detection:

Hardware built–in error detection methods (EDM): They are provided by the hardware platform. They include floating point exceptions, illegal instructions, etc.

Memory access behaviour: They are also provided by the hardware layer. They detect deviations from the proper memory access behaviour.

Node–level error detection: Consists of watchdog timers, by which each node expects *I am alive* messages from neighbouring nodes at fixed intervals. Absence of message is detected.

Communication–level error detection: Parity–bit tests, detection of disconnected links (for example) at hardware level, plus eventual EDMs at high levels.

Control–flow monitoring: Consists of two mechanisms: assigned signature monitoring, ASM, and error capturing instructions, ECI. The first is based on the insertion of milestones on the code at compilation stage, which are tracked in run–time for detecting deviations. The second consists in the insertion of trap instructions in the code, at points where they should never be executed. Their execution in run–time implies an error.

Application–level error detection: The FTMPS considers two types. First, a set of timers for monitoring the behaviour of the system in time, against a set of pre-established timeouts. Second, ad-hoc methods implemented by the user.

It can be observed that the first four are centered in the detecting hardware faults, while the last two are dedicated to software faults. In particular, the last two are particular cases of a category called behaviour-based error detection, BBED.

BBED is based on a restricted model of the system, built off–line and included in the system. In execution time, the system will check its behaviour against this model, and consider deviations as errors. It has been mentioned above that the FTMPS considers a model of its control–flow and of its application–flow. Other systems could also include models of:

- HW control signal behaviour.
- Reasonableness of results.
- Processor instruction set usage.
- Timing features.

Errors are *singularities* considered by the perceptive processes of the homeostatic functions of the system. Faults stand for the *referents*. In fact, "*Error detection* is the phase in which the presence of a fault is deduced by detecting an error in the state of some subsystem" [Jal94, p.9].

We may observe, in case of BBED, that the restricted self–model that the system uses is an example of a set of *potentially instantiated quantities*. The actual *model of the system* is given by the singularities: errors and other parameters measured across the system (such as the four proposed above) and the derived referents (including the considered faults.)

14.4 Damage Confinement

Error code propagation goes from lower to upper levels. Application processors in the D-net execute node–level error detection through a testing module. This module executes watchdog timer processes (the so called *I am alive* messages) which detect inactivity in neighbouring processors, periodic diagnostic routines and error capturing instructions, ECI.

In the event that a neighbouring processor is detected faulty, the code is propagated to the control processor for local diagnosis. The code is in turn, propagated to the global diagnosis module in the host. This module proceeds to any of three actions: terminating the application, distributing local diagnosis results or processing diagnostic information.

Testing modules are an example of *proximal information processors*. Cognitive processing is performed in the local diagnosis modules —i.e. the local control processors. The Global diagnosis processor performs *cognitive processing* as well as *core processing* for decision–taking, and *efferent* functions (the three actions listed above.)

14.5 Error Recovery and Continued Service

FTMPS implements two kinds of error recovery and continued service:

Reconfiguration: Permanent errors require a reconfiguration of the system before the application may continue. It assumes the system has spare or *undamaged* resources for mapping a virtually perfect system into an injured one, and masking the errors. It consists in two phases:

1. Rerouting: This is the calculation of new routes for communicating the active nodes after a faulty one has been identified. In this case (a component has failed permanently) routing tables must be reprogrammed so that faulty entities are detoured.

   The original routing is based on the regular structure of the hardware layer, and tries to make the routing following this structure. Faulty nodes, however, destroy this regularity, so routing tables for communications must be recalculated and adapted. It must be remarked that the number of available communication channels can be a limiting factor in adapting an injured system to a newly occurred fault.

2. Remapping: In the case that the current partition is not useful any more due to a fault, a new partition must be found for running the application. The repartitioning algorithm must find sufficient working processors for the application. The algorithm will try to remap the application in the spare processors of the original partition first. If there do not exist enough spare resources, the application must be remapped into active ones, making the overall performance of the system decrease (graceful degradation.) In this case, the user is given a tool for global repartitioning of the system.

Checkpointing and Rollback: Periodically, a consistent view of the application is saved onto secondary storage (checkpointing). After a fault has occurred, the application is then restarted (rollback) from the most recent set of checkpoints stored, which is called a recovery-line. This avoids starting the application from scratch.

It must be noticed that restarting a multi–process application requires having a set of consistent checkpoints (recovery-line,) for setting the state of all processes at the point of rollback. During operation, a controller keeps track of the checkpoints that are saved, and agrees with other controllers about which recovery-lines become complete and valid. When performing rollback, the controllers determine which recovery-line is the most recent one.

As it has been mentioned previously, rerouting and remapping are two examples of *grounding* mechanisms. They represent a basic form of *functional decomposition*.

We might realize that these processes represent *functional decomposition* of *level 1*. They modify the *grounding* of the functions which realize the user application in order to adapt it to a new situation. The *algorithm* of the functions remains unaltered, as the system has no knowledge either for generating new algorithms or for selecting different ones. Rerouting and remapping, however, represent two degrees of functional decomposition, as the first stands for a redistribution of couplings, and the second also for a redistribution of algorithms among the hardware resources. Major remappings are left to the user.

We may observe that the FTMPS does not perform *objective configuration*. Specifications are not modified by the system mechanisms.

Part V

Epilogue

Chapter 15

Discussion, Conclusions and Future Work

This work provides a basic framework for general, autonomous systems and for perceptive subsystems of them. Accordingly, the work provides a broad view covering many aspects of systems, aiming to build a global understanding.

The framework is formed by highly abstract concepts inherited or generalized from previous studies and experience. Many of these concepts have not been integrated within a unified vision of systems before, either because they were originated in very different disciplines, or because, due to disparity in abstraction or field of application, they seemed totally independent within their original domains.

15.1 Revisiting the Objectives of the Work

I. Generality was a first objective of this work. It has also proved a need at its conclusion. The heterogeneity of systems requires general conceptualization in order to be able to perform a unified analysis. We understand that this objective has been achieved, due in part to the adoption of the Theory of General Systems as a theoretical background.

II. Obtaining concepts, principles and relations of application to system engineering was a second objective. We conclude that the resulting ontology enhances current knowledge by identifying major notions and principles underlying autonomous systems and their operation. Engineering profits from this by gaining a new perspective on systems which allows better comprehension. However, we consider that the objective is achieved only in part, for the applicability of this work is limited as yet:

- Principles of design, specifically stated in the work or indirectly derived from it, allow application at a qualitative level. This might provide the engineer with a sound view of the relation between autonomy, system properties and perception —among other aspects— which might eventually prove useful for overall design or gross analysis, but does not sustain detailed analysis or synthesis.

- System complexity proves a significant issue, given that some of the concepts introduced here may significantly vary their form depending on system complexity, as follows from the mentioned possibility of *degenerated* elements, for example. Theoretical development relating the present ontology to complexity is necessary. Also, a systematic, complexity–dependent methodology of application. A starting point for this can be found in the simplification principles outlined in [Kli01, p.159-170].
- The limitation for the full applicability of this work is not restricted to complexity. Although examples provided illustrate a collection of cases of application, a systematic methodology is not provided, therefore leaving a wide undetermination. It is envisaged that such a methodology should be iterative and progressive, based on a collection of fundamental traits to be initially determined and progressively enhanced, both in analysis and in design. This methodology should specify criteria for establishing fundamental traits and designing iteration steps according to the system environment, resources and objectives. Currently, application is largely left *ad–hoc*.

It must be remarked that a major point for the applicability of this work to artificial systems —analysis and synthesis— is the introduction of *objectives* into the relation of causality inherited from [Kli69]: *properties → behaviour*, giving them explicit theoretical relevance. In fact, this results in an important relation in which the work is based:

$$\text{objectives} \quad \rightarrow \quad \text{properties (organization)} \quad \rightarrow \quad \text{behaviour}$$

Where *objectives* stand for the purposes which *direct* the evolution of the system; explicit or implicit, immediate or long-term, proper to the system or given by the designer.

The work portraits systems and perception including objectives as an *intrinsic*, constitutional aspect of systems, as are mass, length or property in general. We understand that, although the importance of objectives in systems has been largely perceived,[1] this work offers a first attempt to analyze objectives as a fundamental part of general systems, trascending particular architectures and implementations.

It is precisely this novelty that opens another point for further research: the development of general methodologies to systematize objective-oriented design, integrating objectives, organization and behaviour. An example of an objective-

[1] See section 7.1 discussing finality in general systems. Of course, objectives are a critical design factor for all artificial systems: the objectives of the designer are embedded in them. Goal–oriented architectures —see section 5.4— are examples in which the organization of the system is given the capability of managing explicit objectives.

oriented methodology in a particular domain is the widely-spread, root-locus technique for control system design [Oga90].

III. Glossary. The objective of building a glossary of terms of perception emerged during the progress of this work. The attempt made can be consulted in part VI. We understand that this objective is only partially achieved. The glossary provided requires revision of a large number of definitions as well as inclusion of new terms.

However, the experience of attempting it has served to learn several points:

- Including the terms of the present work appropriately requires defining a large number of other, related terms. This may increase the size of the task significantly.
- Defining a term may, in some occasions, require lengthy explanations. The definition of some terms is simplified by adding second meanings, and meanings in different contexts which may allow the reader to refine the notion.
- Including terms of the degree of generality treated in this work makes it more practical to separate context–specific terms from general–scope ones in separate glossaries.

15.2 Future Work

We understand that the concepts and relations introduced in this text will consolidate through sustained and full immersion in the scientific method —see figure 15.1—.

The consolidation of this initial, theoretical proposal stands for two main issues:

- Refinement of scope, precision and relations of concepts.
- Building of a broad application case history.

The second stands for experimentation, upon which to observe conceptual inaccuracies, incompleteness, contradictions, necessary to refine the concepts. It also constitutes the basis knowledge corpus from which to build methodologies of application. We regard this last point as essential.

Bearing these considerations in mind, we envisage the following major lines of work for the immediate future:

Formalization. It is necessary first, to enable systematic engineering, modelling and reasoning with the concepts proposed here. Second, as a tool for refining them, as it provides an unambiguous means of conceptual representation.

Initially, formalization was projected as part of this work. An insight into geometry, mathematics, knowledge representation, software modelling and other topics was given in search for formalization tools. The conclusion was reached that the task itself proved a matter for future research.

As it has been mentioned above, a main objective of this work has been to relate behavioural aspects of systems (external) with purposive and organizational

Figure 15.1: Scientific Method

ones (internal). Accordingly, formal expressions are sought which would make this relation explicit. This is particularly difficult regarding essential concepts introduced in this text. Minimal structure, homogeneity, encapsulation, isotropy of knowledge, scalability, order of objectives, and autonomy are only examples. Although these concepts allow us to better understand systems, they are difficult to express formally, quantitatively and without ambiguity. We have envisaged two potential tools for formalization:

- *Category Theory* [LS97], [Pie91], as a possible tool for modelling and formalization, given its systematic treatment of *sets* and *morphisms*. We must bear in mind that the inspiration of this work, the Theory of General Systems, is based on the *relations* between things more than on the study of the *things* themselves,[2] in line with the identified need for formalizing morphisms.
- *Executable models.* Some tools allow deploying software systems from conceptual models, typically expressed in UML or similar modelling languages. We intend to build models following the ontology proposed in this text applied to particular systems for building a case history and refining concepts.

Research. There exist multiple aspects regarding systems which have been identified here of which we lack knowledge, and which cannot be implemented now, or

[2]See the first chapter of [Kli01] for an explanation of the scope and aim of systems science.

only to a very limited extent. Research is necessary to theorize them, so that they can be included in the ontology initiated here.

A significant example of the need of further knowledge refers to the *principle of scalability*, introduced in p.149. We are unaware of general principles or methodologies of system design —or analysis— for achieving property invariance with scale. We have identified, however, two topics of special interest in this matter: theory of fractals [Man00], and geometry and the study of growth [Coo14], [Ghy83], [Tho61]. In both cases, research is necessary to establish consistent isomorphisms between their frameworks and the concepts proposed here.

Application: Methodologies, Analysis—Synthesis. We consider that applying this work is essential. This implies designing grounding methodologies and experiments. As it was mentioned before, the objective would be to build a broad case history including systems heterogeneous in complexity and nature.

As we have mentioned, executable models are presently in the process of being tested in real systems. A mobile robot, a continuous process plant and a software system will be the three targets considered.

15.3 A Unified Theory of Perception

We can regard the theory proposed in this work as a unified and **unifying** approach to perception. Its generality —a major objective— is in a way, precisely, a strategy for unification.

Accordingly, it is possible to put other approaches to perception in the context of this framework, and see how it can be particularized to each case. In chapter 8 we described them from a neutral perspective. We now develop a short comparative discussion covering the main trends. We regard this as an interesting result of the thesis.

1. Abductive perception. Perhaps the formalization of perception which is closest to this work is that found in [Sha05]. The understanding of the phenomenon is similar in many aspects:

- Point 1 of this thesis >*chapter 9* on the perceptive process and the *fundamental sequence* is basically shared, allowing a certain degree of *proximal information processing* and a phase of *cognitive information processing*.
- The actual role of *singularities* is also identified as not necessarily a description of the external world, but of the state of the sensors (*sensory system* in this work.) This implies the existence of a certain *cognitive equivalence* to be established by the inferential process (*cognitive information processing* in this work.)
- The notion of *umwelt* is also described as part of the process in similar terms to this work >*p.196*.

- It also assumes that perception implies both a *bottom-up* information flow —from proximal information procesing to instantiated referents— and a *top-down* flow —from stages within the *cognitive information processing* phase to the *proximal information processing* phase—.

There are, however, some points of difference:

- This work provides further detail regarding the actual operations of *proximal information processing*: *equalization* and *singularity equivalence*, not analyzed in [Sha05].
- This work develops a framework of general autonomous systems, which provides a detailed description of the context in which perceptive processes exist: nodes, functional structure, objectives, finality, etc.

  Although [Sha05] refers to sensory fusion, which implies multiple perceptive processes, this is only a particular case, which leaves systemic aspects uncovered: relation of perception with core and efferent processes, functional decomposition, directiveness, etc.
- Top-down information flow is assessed only in the particular case of 'expectation'. The term is understood as 'prediction', and it is described as a heuristic mechanism included in the inference carried out in the *cognitive information processing* phase.

  According to this work, however, there exist multiple mechanisms of top-down flow. Implicit perception *>arrow 4, figure 11.2, p.185*, and *re–sensing*, illustrated in case (c) figure 10.6 are examples of this. Also, the influence of higher levels in the functional structure over lower levels.

  The developed context of general autonomous systems developed here also allows to identify other kinds of factors influencing the inferential process, apart from problem–solving oriented heuristics: real-time constraints, resource constraints, coordination constraints, etc., and implicit factors through the *substratal coupling*, *>figure 10.5, p.177; arrow 5 figure 11.2, p.185*.

This work can be considered to follow the major ideas of abductive perception. The similarities with [Sha05] and with notions and views in other works [Roc85], [Roc97] are clear.

However, it is formulated from a wider context including systemic aspects. This allows realizing their actual influence and relevance on the process and achieving a higher degree of generality.

2. **Direct, sense-data and mediated perception.** As it was introduced in 8, *affordances* and *sense-data* are analogous in the assumption of a direct character in perception. This is supported in some contexts by evidence cited in the bibliography of both areas. However, we conclude that (1) they explain specific aspects of perception but lack generality (2) in accordance with this, their scope can be determined in terms of the present work. We shall now attempt this in order to comment further.

- A first approach to representing direct perception in the terms of the present work is shown in figure 15.2. Our notion of *referent* is implicit in direct perception. However, it coincides with the observer system.[3] It can be observed that, according to ecological perception, the perceptive process consists of a unique phase from proximal stimulation to the perception of affordances: surfaces as potential support, [Gib87, p.127], substances as nutrition [Gib87, p.128], etc.

Figure 15.2: Direct Perception in Terms of this Work.

According to direct perception, the sensory systems of animals are intrinsically adapted to perceiving affordances. This is the reason why perception is direct. It means that affordances are perceived exclusively by proximal processing. In terms of the present work, this equals to saying that the stage of *cognitive information processing* proposed here has no bearing in perception. We shall say that cognitive information processing is the identity, i.e. that it yields an identical result to its input. Direct perception is thus represented in case (a) of figure 15.3.

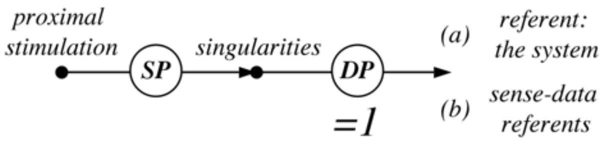


Figure 15.3: Fundamental Sequence of Direct Perception and Sense Data.

We might observe that this is equal to saying that direct perception occurs on —at least— two particular conditions with respect to the general case presented in this work:

  - Cognitive information processing is a unit process. In other words: the represented referent equals the singularities processed by the perceptor.
  - The set of singularities provided by *proximal information processing* are actually meaningful as to what the environment actually affords. This implies that the resources involved are adapted to that purpose: sensory system, relations between quantities, rest of processes in the system, etc.

[3]Note that: (1) The system perceives the *affordances* of the environment. (2) "[Affordances] have to be measured *relative to the animal*" [Gib87, p.127]. In conclusion: the *animal* —i.e.: observer system— is the referent of the perceptive process.

- Regarding *sense-data* perception, we shall assume that it represents a more general theory than *direct perception*. The main points of difference between both are as follows:

  1. Sense-data theory admits that sense-data are mind-dependent, while ecological perception claims that affordances are exclusively environment-dependent.[4]
  2. Sense-data theory admits that the meaning of sense-data may not be referred to the system, but to the intrinsic properties of the objects in the environment: a red tomato being red, an orange being round, etc.

  These differences make sense-data theory more general because it explains perception to a higher level of abstraction —intrinsic properties of the environment, regardless the observer—, and it permits explaining the influence of the observer itself in its own perception: past experience, memory, etc.

  Sense-data theory may be expressed in the terms of this work as in case (b) of figure 15.3. It must be remarked that: the *cognitive information processing* phase is also an identity, and that the referents of perception are not required to be system-oriented as in the case of direct perception.

In this light, we may raise the following points:

1. Both approaches impose a unit *cognitive information processing* phase in perception. This implies that the *proximal information processing* phase necessarily has to be adapted to the process *referents*. In other words, the sensory system must be *specific* to the referents: the resources on which it is embodied and the singularities it considers.

   The range of perceivable referents is restricted by the specificity of the sensory systems. If a sensory system would be too specific, new or modified referents could not be perceived.

2. Both theories are largely based on physical attributes of the environment. Perception of abstract referents based on abstract or conceptual *singularities* is not accounted for.

   The ecological approach would categorize this kind of processing as *second-hand* or conventional [Gib65]. However, it is clear that first-hand and second-hand processing are related and mutually influenced. Also, that second-hand processing has effects in terms of physiological response and activation of brain areas which in many cases are undistinguishable from first-hand processing. The relation between first- and second-hand processing is not accounted for. This aspect has been systematically treated throughout this text, and specifically in section 10.3, p.176.

3. *Affordances* as defined in the ecological approach [Gib66] [Gib87] are referred to aspects such as *support* and *nurishment*, which answer finally to system

[4] Although, as we have seen, they have to be understood as system-dependent.

survival. In the terms of this work, this would be regarded as a *root objective* of the system.

However: (1) a system, in the general case, may have more root objectives apart from survival. (2) *Survival* may not necessarily be a root objective in all systems, especially in artificial ones.

4. *Root objectives*, as it was developed in part II, are the most abstract and longer time-scope objectives of the system. They are realized by a structure of *intermediate objectives* which might differ significantly from them. This structure is adapted to shorter time-scopes and levels of abstraction, according to the instantaneous requirements imposed by the environment and the capacities of the system. Therefore, the higher the degree of autonomy of the system, the lower the degree of specificity of the system should be —according to the principle of minimal structure—.

In conclusion, we may say that affordances are a more particular formulation than sense-data, as they impose a more restrictive view on perception. The lack of generality of both approaches leaves multiple aspects of perception uncovered, especially regarding perceived objects.

The direct character they attribute to perception does not allow explaining coordination and other forms of mutual depencence between perceptive processes in complex systems, where multiple processes might be taking place concurrently.

However, it must be remarked, against purely symbolic notions of perception, that in the general case a phase of *proximal information processing* must be contemplated, although it will, in general, be dependent on the operation of the rest of the system. Proximal information processing can adopt high degrees of development including equalization and singularity equivalence functions.

3. Gestalt perception. The present framework has relation with Gestalt perception in key aspects:

Singularities stand for relations between the values of sensory system quantities. In turn, *cognitive equivalence functions*, ϵ, *>section 11.3.2*, stand for relations between singularities.

This framework is therefore based in the concept of relation among parts, following the Gestalt inspiration.

- Gestalt assumed that perception was concentrated on the analysis of some relations such as symmetry. However, this work imposes no constraints in the relations that a perceptive process might consider as singularities. A review of the literature regarding low-level perception in biological systems shows sufficient evidence as to the heterogeneous nature of singularities, that no restriction can be imposed on the notion.

  Examples of singularities in biological systems are: spatial proximity/continuity/symmetry of values as in object recognition, proximity/continuity of values in time, as in event-following, discontinuity of values in time, as in attention shift, and frequency spectrum patterns as in voice recognition. These examples show intrinsic differences of kind among the particular forms that singularities might adopt.

- This work explains perception within the broader notion of *node*, which in turn is within the larger context of the *functional structure*. This implies that perception must answer to more criteria than *optimality*, as Gestalt postulated. Namely: real-time constraints, coordination constraints, resource constraints and finality. The existence of these constraints explains why singularities might present such different natures, and why their interpretation is not necessarily optimal in real perceptive systems.

4. **Marr Theory of Vision.** This work has multiple ideas in common with the Marr theory of vision:

   - Both are aware of the duality between representation and processing. In this work this duality resides in the role of the *referents* in perception and in the role of *implicit perception* in the system.

     Referents largely determine the *point of view* of a perceptive process *>section 11.3.1, p.191*, and therefore influence the intermediate phases including the implicit perception derived. Implicit perception stands for the influence of the process of perception over the rest of the system. Referents also influence the *percepts* that the process will produce, and consequently, the derived cognitive processes.

   - Both distinguish a qualitative difference between the analysis of the *primal sketch* (*proximal stimulation*) and the rest of perceptive processes. However, from the point of view of this work, there is no qualitative, fundamental difference between Marr's $2\frac{1}{2}$ and 3-D models.

     According to Marr, the $2\frac{1}{2}$ and 3-D models differ in their point of view. The first is centered in the system and the second is neutral. This work assumes that each perceptive process has its own *point of view*, *>section 11.3.1, p.191*. The main aspect that defines the point of view is the *referent* of the process. Therefore, system-centered perception or neutrally-objected perception answer mainly to different referents, but there is no qualitative or fundamental difference.

   - There is another point of analogy with Marr's vision: the distinction between *computational*, *algorithm* and *implementational* levels of vision. These levels correspond —conceptually— to the functional levels of systems identified in section 7.4.2: *functional* —node level—, *algorithm* and *grounded function*. In fact, the importance of these levels in system adaptivity was introduced with the concept of *functional decomposition*, p.145.

15.4 Major Novelties of this Work

We may conclude from the previous comments that the major novelties of this work derive from the generality of its perspective.

Usually, other approaches to perception part from a particular area of knowledge, concentrated on a specific part of the problem, from which generality is achieved by progressive steps. This explains why each approach has usually identified isolated phenomena.

This approach has adopted the opposite line, first developing a notion of autonomous system: directiveness, finality, objectives, organization and behaviour, integrated within a unified view, and complemented with a procedural description of the system —*node structure*, *node elements*—. This notion forms a complete context for perception and allows:

- Establishing a comprehensive view of perception: (1) form: multiple, distributed processes and (2) operational —node— and grounding context: constraints of coordination, communication, resource allocation, substratal dependence.
- Identifying multiple implications, including for example: points of influence of perception on the system and vice-versa, non-ideal processes in perception (for example non-optimal criteria of recognition —opposite the Gestalt approach,—) perception in time —perceptive dynamics, >*section 10.3*—.
- Building a broad collection of related phenomena to be explained in perception: the role of memory, the influence of the rest of the system in a perceptive process, heuristics, emotions, etc. Other approaches of perception focus only on some of them.

In conclusion, generality provides a unified and complete collection of implications, processes and phenomena to which perception must answer. It also shows —qualitatively in this work— the relative importance of each as to the system and to each other.

Part VI

Reference

Chapter 16

Glossary of Cognitive Systems

keys

>˜	reference to entry ˜
abrv.	abbreviation
ai.	artificial intelligence
biol.	biology
cogsci.	(cognitive) (neuro-) science
eng.	engineering
gst.	general systems (theory)
phil.	philosophy
psychol.	psychology
rob.	robotics

A

abstraction: ***1.*** process of conceiving systems or situations by progressively omitting their more frequently changing aspects or the aspects more related to physical quantities ***2.*** notion of a system or situation which only considers structural aspects ***3.*** *eng. for comparing processes* indicates longer temporal horizon of a process (lower temporal resolution.)

abstraction spectrum: *gst.* number and distribution of abstraction levels of the functions of a system >functional structure.

ACT-R: *cognitive architecture* for simulating and understanding human cognition focused on how people organize knowledge and produce intelligent behaviour. http://act-r.psy.cmu.edu/

action: ***1.*** *eng.* activity of a system's outputs ***2.*** *physical action in rob.* [Mey00, p.4] complete set of agent motion (or behaviors) that are developed by actuators of the agent and are sensed by the agent as changes in the external world.

activation of objectives: >*activity of an objective* ***1.*** *gst.* process by which an objective instantiation is applied to the system becoming cause of behaviour ***2.*** *gst.* process by which a >*finality* is implemented in a system.

activity: *of a system [Kli69, p.41].* The ensemble of the variations in time of all the quantities under consideration at a given resolution level.

activity of an objective: *gst.* ***1.*** period of time during which the system is directed toward the objective ***2.*** characteristics of the system caused by the objective during that period ***3.*** values of the quantities affected by the functions associated to the objective during that period.

actuator: ***1.*** *gst.* conceptualization referring to efferent aspects of a node function ***2.*** *eng.* device or set of devices which transform an electric signal into a physical magnitude such as: motor, servovalve, etc.

adaptation: *cogsci.* generation and/or change of the system organization toward achieving a particular objective which makes possible or improves the possibility of succeding with respect to the previous organization >*finality (2b)*.

adaptivity: capacity of >*adaptation*.

agent: ***1.*** system *emphasizing that its function is part of a more complex function:* subsystem ⇒ node ***2.*** *ai. [Mey00, p.31] sometimes autonomous ˜ or intelligent ˜* a system that can probably sense, reason and is intended to act. ***3.*** *eng.* [Alb91, p.4] set of computational elements that plan and control the execution of jobs, correcting for errors and perturbations along the way.

ARTIS: *cognitive architecture* for intelligent, multi-agent, hard real-time control systems http:// www.upv.es/sma/web/rta.htm

ATLANTIS: *cognitive architecture* three-layered architecture integrating behaviour-based and deliberative architectures http:// www.flownet.com/ gat/papers/aaai92.pdf

attractor: *eng. when analyzing the state dynamics of a system* stable state.

autonomy: ***1.*** *gst.* combined form of degree of interdependence and functional capacity of a system at a certain instant of time ***2.*** [Mey00, p.31] an ability to generate one's own purposes without any instruction from outside ⇒ *ability of the system to generate its own goals without external instructions >decisional autonomy* ***3.*** *rob.* capacity to operate without human intervention >*operational autonomy* ***4.*** internal cohesion ***5.*** *biol.* homeostasis.
–absolute autonomy: ideal case in which degree of interdependence is total (there do not exist independent quantities) and functional capacity is infinite.
–decisional autonomy: *eng.* capacity of a system to generate its own objectives.
–ideal autonomy: >*absolute autonomy*.
–null autonomy: all system quantities are independent.
–operational autonomy: capacity of a system to operate without external intervention *in some contexts,* without human intervention.
–total autonomy: autonomy of a system having a single independent quantity affecting only behaviour of order 0.

awareness:

B

bandwidth: *of a system gst.* maximum possible frequency of change in a system's behaviour of order 0.

BB1/AIS: *cognitive architecture* multi–agent, blackboard architecture; principal researcher: Barbara Hayes-Roth; http:// www.ksl.stanford.edu/projects/AIS/

behaviour: Particular time-invariant relation specified for a set of quantities and a resolution level, and based on samples of a certain pattern.
–permanent (real) behaviour: The set of all local relations.
–relatively permanent behaviour: *(known) behaviour* Set of all local relations of a particular activity: relative relation.
–temporary behaviour: Local relation corresponding to a distinct section of a particular activity.

C

causal relation: *gst.* ***1.*** a set of ordered pairs (cause,result) ***2.*** a special type of relation whereby each of the dependent quantities can be expressed explicitly and uniquely as a function of the other quantities, and the independent quantities cannot be expressed explicitly, or their explicit expression is ambiguous [Kli69, p.62].

characteristic: ***1.*** pattern of behaviour of a system which are particular to it and, consequently, help distinguish it from other systems ***2.*** let $A = \{a_0, a_1, \ldots, a_u\}$ be the set of elements of a system and the environment (a_0), $B = \{b_1, \ldots, b_u\}$ their respective permanent behaviours, and C the set formed by all elements $c_{ij} = c_{ji} = b_i \cap b_j, \quad i \neq j$. The set C is called characteristics [Kli69, p.54].

–constitutive characteristics: *gst.* [vB69, p.54] dependent on the specific relations of a system i.e. dependent on its environment.

–summative characteristics: *gst.* [vB69, p.54] those of an element which are the same insde a system than out.

CLARION: *cognitive architecture* for modelling cognitive processes. Key aspects: integration, representation, implicit-explicit interaction http:// www.cecs.missouri.edu/˜rsun/clarion.html

COGAFF: *cognitive architecture* offering models of integration for aspects derived from human cognition such as emotion and consciousness http://www.cs.bham.ac.uk/˜axs/cogaff/

COGENT: *cognitive architecture* for amplifying human perception in complex environments. Capacities for multi-level decision taking, information processing, situation assessment [DG00] http:// delivery.acm.org/ 10.1145/340000/337564/p443-das.pdf

COGNET: framework for analyzing and modelling human behavioral and cognitive processes in real-time, multi-tasking environments; metacognition [ZJ00] http:// www.cognitiveagent.com/

cognition: ***1.*** [Gra96] the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses ***2.*** [Gra96] a result of this; a percept [-ion], sensation, notion or intuition ***3.*** process of perception ***4.*** process of perception or percept, understood within a certain >*finality* for it.

cognitive integral component: >*integral component*.

cognitive functional components: >*functional components*

cognitive point of view: *when analyzing a node* analysis of the node function by decomposing it into its cognitive components: afferent, efferent and deliberative. *complementary to >functional point of view >functional components*.

complexity: ***1.*** qualitative valuation of the number and form of the traits of a system, relative to that of the observer ***2.*** *as in 'very complex'* equality or superiority of the traits of a system with respect to its observer or to another system in: number, form, abstraction.

consciousness: ***1.*** *ai. inspired from [Mey00, p.17]* process that provides the system with a view of the self in the context of the immediate environment ***2.*** *cogsci.* facility for accessing, disseminating, and exchanging information, and for exercising global coordination and control [Baa97, p.7] ***3.*** *cogsci.* inspired from [Tay99, p.345] process for using traces from the past to clarify what comes afterward in achieving the goals of the system ***4.*** *psychol.* phenomenologically, extracted from [Den91, p.45] (a) experiences of the external world *ie: exteroception & proprioception* (b) experiences from the internal world *ie: fantasies, daydreams, etc* (c) experiences of emotion or affect >*emotion*.

constraint: ***1.*** *gst.* scenario, situation or other aspect which limits the range of variation of a quantity or set of quantities ***2.*** *gst.* time-invariant relation of system quantities in which there appear independent quantities ***3.*** *gst.* time-invariant relation.

context: *when referring to a system or element* reference to a particular configuration in relation to its environment; may also refer to a state of the environment, the system, or of both under specific conditions of time (at a particular instant or time interval.)

control: ***1.*** process of generating quantity values in order to direct a system or subsystem toward an objective ***2.*** generation of objectives of lower order ***3.*** *adj. gst.* classification of system quantities in inputs and outputs *eg: the control of the system is known* [Kli69, p.65] >*neutral*.

coupling: ***1.*** *gst. of elements* the set formed by all common external quantities of the elements ***2.*** influence or relation between two systems or parts.
–hypothetic coupling: *gst.* coupling which exists during a particular activity of the system, but may not exist in others.
–real coupling: *gst.* coupling that exists over the entire time interval of any activity of the system, and therefore is understood to be independent of a particular situation: proper to the system or *real*.

D

deactivation of an objective: ***1.*** *gst.* process of change in a system after which its evolution will not be directed to the objective ***2.*** *gst.* overriding of its instantiation.

degree of interdependence: ***1.*** qualitative valuation of the absence of influence of the environment over a system ***2.*** *gst.* notion equivalent to the number of degrees of freedom a system has in a given context at a particular instant of time. One of the factors for autonomy of a system >*functional capacity*.

deliberation: ***1.*** creative thought ***2.*** *gst.* productive thought as in [May86, p.56]: process of generating *(ie: producing)* a new solution to a problem.

deliberator: ***1.*** *gst. abstraction* deliberative components of a node: function and resources ***2.*** *gst.* deliberative functions and resources of a system considered as one *(ie: aggregation of smaller deliberative component functions)*.

differential process: *gst.* ideal process of infinitely short duration and content, resulting from an infinite abstraction spectrum or continuous abstraction spectrum.

disturbance: *eng.* environmental factor which affects the output of a system and is not a system input >*perturbance*.

dynamic knowledge: *gst. in a node* referers to the part of the knowledge of the system required by the node function during execution (eg. as a dynamic resource,) produced by the afferent component of the node during execution, and used internally by the other components, and also to dynamic references to system knowledge *(external to the node)* >*working memory*.

E

element: *of a system* ***1.*** smaller part of a system: subsystem ***2.*** *gst.* subset of the system quantities, distinguished so as to express cohesion of physical properties or other relations between element quantities. The system can be expressed as a set of elements.

emergence: ***1.*** appearance of behaviours in a system, resulting from its interaction with the environment (especially with other systems in the environment) which is not explained by the quantities chosen by the observer for representing the system. There are two reasons for emergence: (a) the system is open but has been modelled as closed (b) there exist independent internal quantities ***2.*** *in some contexts psychol. phil. cogsci.* phenomenon by which a system may exhibit behaviours in the presence or combination with other systems, which cannot be inferred by its study in isolation.

emotion: ***1.*** *cogsci. after [Dam00, p.79]* combination of (a) a neural pattern (recognition of a certain object in memory), (b) a body state and (c) a feeling of the previous, aroused by either the process of perception in general or by reminding. Neural pattern (mental image) and body states are reactions to certain stimuli associated to the distal stimulus of the object. ***2.*** *cogsci. after [Dam00, p.53]* biological function to (a) produce a specific reaction to an inducing

situation, (eg. run away) and (b) regulation of the internal state of the system to prepare it for the specific reaction (eg. increase leg bloodflow.) In humans these reactions may be tempered by by cognitive levels.
–background emotions: *cogsci.* being well, calm, tense, etc. Triggered by internal state, by interaction with the environment or both [Ber06, p.19-20].
–primary/universal emotions: *cogsci. after [Dam00, p.50]* happiness, sadness, fear, anger, surprise, disgust. Correspond to the intuitive notion [Ber06, p.19-20].
–secondary/social emotions: *cogsci. after [Dam00, p.50]* embarassment, jealousy, guilt, pride, etc. Result of the individual within a society [Ber06, p.19-20].

Encapsulation, Principle of: *gst.* principle of design which favours decomposing functions in functions of lower level of abstraction so that each interacts with the rest through an explicit interface *ie: minimizing implicit dependencies*. In this way, a component realizing one of these functions can be substituted for another with the same interface and specification.

encoding: *cogsci. in the sense of [New90, p.59]* process by which an external situation or process external to the system is related to an internal state *ie: representing*. *>perception >memory*

ERE–Entropy Reduction Engine: *cognitive architecture* for integrating planning, scheduling and control. Principal researchers: John Bresina and Mark Drummond [DBK91].

environment: *in relation to a certain system or object* part of the universe which is not part of the system.

evolution: *of a system* ***1.*** abstract reference to the change of a system in time, observed in any of its traits, ***2.*** the previous, focused on a particular trait, generally structural.
–dependent evolution: *of a system gst.* evolution of the system due to dependent quantities and relations among them.
–independent evolution: *of a system gst.* evolution of the system due to independent quantities.

Explanation: *gst. of a time-invariant relation* expression of a time-invariant relation in terms of simpler relations.

extensive: *adjective* opposite to *>intensive*.

F

feeling: *cogsci. after [Dam00, p.280]* mental image of a neural pattern corresponding to the changes in the body and brain forming an *>emotion*.

finality: ***1.*** *gst.* [vB69, p.77] usefulness or adequacy of something for a certain purpose. ***2.*** *gst.* [vB69, p.77] *dynamic* Directiveness of some process towards a final state: (a) expressed as if present befavoiur were dependent on that final state, (b) as if the structure would lead to that final state (c) as if present behaviour were determined by the foresight of the goal. ***3.*** Objective.
–equifinality: [vB69, p.79] fact that the same final state can be reached from different initial conditions and in different ways.
–region of equifinality: *gst. eng.* set of states or region of the state space from which the system would evolve toward a same final state.

formal information: ***1.*** *in a node gst.* information necessary for the node operation such as: internal registers, time labels, representation of the associations of the node with the rest of the system, etc., of non-algorithmic content ***2.*** explicit description of structures and structural aspects, *especially* when this information refers to the system or parts of the system that operates with it.

fractal:

function: ***1.*** objective or purpose of a mechanism, process or resource *eg: it has the ˜ of indicating temperature* ***2.*** *gst.* time-invariant relation *ie: used to indicate that the relation is associated to a particular final state*

3a. (partial) process specification dedicated to achieve a particular objective ***3b.*** *gst.* process specification which totally or partially defines the dynamic directiveness of a system towards a final state >*finality–2*.
–atomic function: *gst.* function associated to objectives of level 0.
–intermediate function: *gst.* function associated to intermediate objectives.
–system function: ***1.*** *gst.* function of the system directed to achieving the generative set of objectives. ***2.*** combination of all functions in a system ***3.*** *used as in* >function–1 generative objectives of the system.

functional capacity: ***1.*** intuitive notion of the effectiveness of an algorithm (efficiency, efficacy, performance, etc.) ***2.*** *gst.* state or set of states reachable by a system when executing a certain function under particular environmental conditions *eg: fridge capable of maintaining temperature at* 5^oC *in a room up to* 65^oC ***3.*** *gst.* one of the aspects determining autonomy >*degree of interdependence*.

functional components: *gst.* components of a node: afferent, efferent, deliberative and integral.

functional content: *gst.* algorithm.

functional generator: *gst.* [Kli69, p.157] element of a system for generating instantaneous values of the output and/or internal variables by both the instantaneous values of the involved variables and the data stored from the past >memory.

functional point of view: *gst. when analyzing a node* analysis of a node function relatively to the functional structure and the objective structure: order, dependencies, resources *complementary to >cognitive point of view*.

functional space:

functional structure: *gst.* ***1.*** topology of dependencies, hierarchies and all associations among the functions of a system. Important aspects of the functional structure are: (a) abstraction spectrum (b) dependencies among functions. ***2.*** idem *emphasizing the correspondence to the objective structure.*
–functional content adaptivity: *gst.* capacity for redefining functional content dynamically.
–functional structure adaptivity: *gst.* capacity for redefining the functional structure (& the objective structure) dynamically; divided into (a) spectral adaptivity and (b) dependence adaptivity.

G

generative set of objectives: *of a system* ***1.*** ultimate purpose/finality of a system ***2.*** objectives of the highest level of abstraction in a system ***3.*** *gst.* objective or set of objectives which cause the real structure of a system.

generation of an objective: ***1.*** process of creating a new objective as a decomposition of one of a higher level of abstraction ***2.*** *gst.* process by which the >*representation of an objective* is created: (a) essence generation (b) code generation and (c) instantiation.
–code generation / coding: particularization of the abstract idea represented by the essence for a specific class of systems.
–essence generation: *gst.* expression of a solution to a problem in problem-contextual terms *ie: regardless its belonging to a specific system*.
–instantiation: *gst.* particularization of the code of an objective for a particular system at a particular instant of time.

goal: ***1.*** a particular objective ***2.*** *in some contexts as [Mey00] and [Alb91]* generative set of objectives, *also referred to as ultimate ˜*.

H

homeostasis: *biol. after [Can39] in [FMD05, p.261]* stable states that are reached at any moment by the physiological processes that work in the living organism.

Homogeneity, Principle of: *gst.* design principle for decomposing a system into elements (or functions into subfunctions), which favours compatibility of interaction among all resulting elements. *eng.* Practical restrictions may yield incompatible elements. In this case, the principle favours creating a middle element with no algorithm for element integration: wrappers, interfaces, protocols, etc.

I

integral component: *gst.* *of a node* conceptualization: resources and processes for composing afferent, efferent and deliberative components *>functional components*, *>formal information*.

intelligence: ***1.*** *rob.* [Alb91, p.474] the ability to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal. ***2.*** *rob.* [Mey00, p.2] the faculty of an agent that allows to deal with knowledge and to achieve the externally measurable success under a particular goal. ***3.*** *cogsci.* capacity to use knowledge to acquire, organize and apply knowledge *>rationality*, *>perception*.
–intelligent behaviour: [Mey00, p.10] characterized by flexible and creative pursuit of endogenously defined goals.

intensive: *adjective* property of being intrinsic instead of arising from accumulation or sum; opposite to extensive.

interface: ***1.*** set of inputs and outputs of a system, through which it interacts with other systems, and all associated specifications for that interaction such as protocols, resolution levels, range ***2.*** idem but emphasizing that the ˜ has been specifically designed and that is is the only means of interaction of the system with its environment ***3.*** *gst.* boundary of a closed system.

isomorphism: *in science* [vB69, p.80] analogy between two or more different phenomena and/or theoretical explanations for it.

Isotropy, Principle of: *gst.* principle of design which favours modes of representation of infinite resolution in contrast to schematic or purpose-oriented representation.

K

knowledge: ***1.*** contents of the system memory ***2.*** *gst.* contents of system memory and functional capacity of a system ***3.*** *rob.* [Mey00, p.2] *(ai)* collection and organization of information units of an agent.

L

learning: ***1.*** process of cognition, perception, rationality ***2.*** *psychol.* processes of categorical thinking, problem solving and memorization of results ***3.*** *rob.* [Mey00, p.2] [Alb91] recording experiences and deriving from them new sets of rules that suggest how the system should act under particular circumstances (in a particular situation and under particular goal) ***4.*** *cogsci.* automation of tasks (chunking) so that they become reactive/unconscious as in [FKID03, MA05, New90].

M

marginal interdependence:

memory: *gst.* element of a system for storing past data *>functional generator* [Kli69, p.157].
–autobiographical memory: memory of events and topics related to the system's own evolution [Mat05, p.129].
–long-term memory: *cogsci.* adapted from [Mat05, p.129] set of memories for past experiences and information accumulated over a lifetime, usually divided *psychol.* in: (a) *>episodic memory*, (b) *>semantic memory* and (c) *>procedural memory*. *>encoding*
–episodic memory: *psychol.* from [Mat05, p.129] memory for events happened to the system.

–semantic memory: *psychol.* from [Mat05, p.129] description of the organized knowledge of the system about the world including words and factual information.

–procedural memory: *psychol.* from [Mat05, p.129] memory for methods of carrying out actions.

–short-term memory: *>working memory*

–working memory: *cogsci.* adapted from [Mat05, p.99] the brief, immediate memory for material currently being processed and for coordination of ongoing mental processes.

metaperception:

Minimal Structure, Principle of: *gst.* design principle which favours minimizing functional structure or making it as adaptive as possible by minimizing designed structural constraints.

modes of an objective: *gst.* two: ***a.*** explicit, when the objective has an abstract representation and ***b.*** implicit, when it has only a real part.

module: ***1.*** *eng.* system with an interface as in *>interface–2* ***2.*** subsystem.

N

natural system: *>autonomous system*

neutral system: *gst.* when the *>control* of the system is not known *>control–3*.

node: ***1.*** *eng. rob.* agent within a network ***2.*** *gst.* a function and the associated resources ***3.*** *gst.* idem during execution time ***4.*** *gst.* set formed by the integral and functional components.

O

object: ***1.*** *gst.* system ***2.*** entity known to a system ***3.*** *cogsci. psychol. rob.* entity identified by a system in its environment ***4.*** *rob.* physical ˜ in a robotic environment.

objective: ***1.*** finality of a system or part of a system ***2.*** *gst.* specification of a desired configuration of a system relative to its environment *>finality–2*. *categories of objective components:* (a) target specification: describing the objective's desired final state and (b) procedure of reconfiguration: a specification of a dynamic sequence for achieving the desired final state (*>finality, setpoint, open objective, closed objective*). *parts of an objective:* (a) essence (b) code (c) instantiation *>objective in abstract form, objective in real form*.

–abstract objective: *>objective in abstract form*

–active objective: *gst.* objective that is currently causung behaviour in the system *>finality–2*, *>inactive objective*. An objective ceases to be active when either (a) it concludes: final state is reached (b) it is deactivated.

–activity of an objective: ***1.*** period of time during which an objective is active ***2.*** changes in the system caused by an objective ***3.*** *gst.* properties of the system derived from an objective.

–dynamic activity of an objective: *gst.* activity of an objective which is directed by the *>procedure of reconfiguration* which form part of the objective specification.

–static activity: *gst.* activity that results from an objective which does not contain *>procedure of reconfiguration* specifications.

–closed objective: ***1.*** *gst.* objective in which the goal and a procedure for achieving it are specified: ***2.*** *gst.* objective containing both target specfication and procedure of reconfiguration components (*>objective*).

–generative objective: *gst.* forming part of the *>generative set of objectives*.

–inactive objective: *gst.* objective which exists as a *representation*, and does not cause changes in the system.

–intermediate objective: ***1.*** *cogsci.* subobjective ***2.*** *gst.* objective of non-maximum level of abstraction: non-generative objective.

–objective in abstract form: ***1.*** an abstract goal >*abstraction*, >*goal* ***2.*** context-independent goal ***3.*** *gst.* objective that does not contain an instantiation part, ie: that is specified in essence, coded, or both >*objective in real form* ***4.*** problem solution.
–objective in real form: *gst.* objective which is particularized for a specific system and context of activation, ie: it is instantiated >*objective in abstract form*.
–open objective: ***1.*** *eng. gst.* process specification designed to achieve a goal which is not explictly mentioned or used in the process *as in open loop control* ***2.*** *gst.* objective containing only >*procedure of reconfiguration* components >*closed objective* ***3.*** *gst.* objective whose instantiation does not contain or use explicitly a representation of the desired goal.
–real objective: >*objective in real form*
–setpoint: ***1.*** a specific target ***2.*** *gst.* objective containing only >*target specification* components >*open objective, closed objective*.

order: ***1.*** *gst.* relative level of abstraction within the system objective or functional structures ***2.*** *gst.* numerical value expressing it (between 0–most specific and 1–most abstract) ***3.*** *gst.* the function to calculate that value.

objective structure: *gst.* set of dependencies, couplings, priorities and all types of associations existing among a set of objectives *usually:* the structure of system objectives.

organization: ***1.*** the way in which a set of entities are arranged with respect to each other ***2.*** *gst. of a system* causes which produce the behaviour of the system *ie:* dependencies between system elements, nature of the elements, etc. which cause the system to behave in a particular way.
–constant part of the organization: >*structure*
–variable part of the organization: >*program*

P

percept: >*perception*

perception: ***1.*** [FMD05, p.80] *psychol. etymol.* capture through the senses carried out by organisms ***2.*** *gst.* perception is the establishment of a relation between an entity (*perceptor*) and its environment (>*perceptive environment*). As a result of this process, the perceptor changes, reflecting the new relation. These changes may be conceived as an entity on their own, coded by the system, which will be called *percept* >*encoding, memory, perceptive environment, perceptor, system environment*.

perceptive environment (PE): *gst.* considering the perceptor as a system, its environment *note that (a) in general* $PE \cap SE \neq \emptyset$ (>*system environment*) *and (b) PE is not equivalent to* >*umwelt/reachable/immediate environment*.

perceptor: ***1.*** system that perceives ***2.*** *gst.* part of a system which carries out perception at a certain instant ***3.*** *gst.* afferent functional component ***4.*** *gst. conceptualization:* aggregation of all afferent components in a system.

perturbance/perturbation: ***1.*** *eng.* aspect, property or state of a system which is not normal (perturbed state) due to a >*disturbance* ***2.*** combination of disturbance and perturbed state.

physical: ***1.*** materials, objects, products, or environments that exist tangibly in space ***2.*** mechanical ***3.*** *gst.* measurable.

point of view: *gst. for analyzing a system* set of space-time resolutions, quantities and time-invariant relations used or followed to describe a certain part of the universe called >*system*, which defines regions of space, instants of time and rules to be followed in the analysis [Kli69, p.31].

procedural information: ***1.*** *psychol. cogsci.* knowledge which refers to the mode of achieving an objective ***2.*** *gst. in a node* algorithms or knowledge of the system that can be employed to specify >*functional content*.

procedural knowledge: >*procedural information*.

procedural set: *gst.* set of all processes taking place in the system at a certain instant of time and their mutual dependencies.

PRODIGY: *cognitive architecture* for planning and learning: automatic abstraction, experimentation, explanation–based learning http://www.cs.cmu.edu/ afs/ cs.cmu.edu/ project/ prodigy/ Web/ prodigy-home.html

program: ***1.*** *eng.* process specification when not being executed ***2.*** *gst. of a system* variable part of the organization of a system [Kli69, p.44], >*organization, structure*.
–complete program: *gst.* instantaneous state with the set of all other states of the system, and the set of all transitions from the instantaneous state to all other states of the system in time [Kli69, p.45].
–subprogram: *gst.* instantaneous state with a nonempty subset of the set of all other states of the system, and a nonempty subset of the set of all from the instantanoeus state to all other states under consideration in time [Kli69, p.45].
–instantaneous program: *gst.* instantaneous state with the transitions from this state [Kli69, p.45].
–state-transition structure: *gst.* constant part of the complete program, formed by the complete set of states and the complete set of transitions between the states, *abrv.* ST-structure. *types:* real state-transition structure, hypothetical state-transition structure [Kli69, p.46].

Q

qualia:

quantity: *gst.* an observed attribute of a system [Kli69, p.37] *ie:* a system is studied by measuring a set of quantities and analyzing their relations.
–abstract quantity: ***1.*** *gst.* [Kli69, p.280] quantity whose values are defined ***2.*** *gst.* non-measurable quantity.
–conceptual quantity: >*abstract quantity*
–external quantities: observed quantities of the system [Kli69, p.44].
–internal quantities: not-observed, mediatory quantities [Kli69, p.44].
–dependent quantity: quantity which is produced by the system, derived from the independent quantities and from the properties of the system [Kli69, p.61].
–input quantity: ***1.*** *gst.* in engineering, quantity which is produced by the environment, *adapted from [Kli69, p.65]*. ***2.*** *gst.* when the input quantity is observed.
–independent quantity: *gst.* quantity which is produced by the environment *ie: it is independent of the system*, and is cause of events in the system [Kli69, p.61].
–output quantity: ***1.*** *gst.* in engineering, quantity that is produced by the system, [Kli69, p.65]. ***2.*** *gst.* when the output quantity is observed.
–physical quantity: [Kli69, p.69] quantity that is measurable.
–principal quantity:

R

rationality: *cogsci. ai. refs.* [Mey00, p.5], [Sim90, p.31,45] ***1.*** capacity of a system to apply knowledge to its own knowledge, metacognition ***2.*** *more specific* logical traceability of behaviours produced by intelligence, by which a relation can be established between the behaviour, the state of the system, the state of the environment and a single or a set of objectives active in the system.

reasonableness: >*rationality*

region of realization: *of a certain objective gst.* ***1.*** region of equifinality *in the system's state space* that leads to a particular objective, and which is possible for the system to reach from its current state >*finality*. ***2.*** intersection between the >*ideal region of realization* and the >*zero region*.

–ideal region of realization: *of an objective gst.* set of system states that realize or direct the system to realizing a particular objective (>*finality (2 a& b)*), ie. states of equifinality >*finality*.

–zero region: ***1.*** *gst.* region of the system state space determined by the zero coupling, formed by the possible states for the system at a certain instant under certain conditions ***2.*** *gst.* instantaneous program.

reference: ***1.*** >*objective–setpoint* ***2.*** mention or citation to an entity.

representation of an objective: *gst.*

resolution level: *gst.* sets of values of all the observed or given quantities to be taken into consideration, together with a set of those time instants at which we want and are able to obtain the corresponding values of the quantities [Kli69, p.40].

–space-time resolution level: [Kli69, p.38] definition of the accuracy and frequency of observations, given a space-time specification.

*–**space specification:*** *gst.* [Kli69, p.37] specification of the points in space where quantities have to be observed.

*–**time specification:*** *gst.* [Kli69, p.37] specification of the instant in which observations start.

resource: part of a system dedicated to the realization of an objective.

–active resource: *gst.* part of the system referenced in a function definition.

–extensive resource: ***1.*** *gst.* resource which can be replicated and instantiated more than once, oppositely to an >*intensive resource* ***2.*** *gst.* abstract resource.

–intensive resource: *gst.* resource of a system which, due to the presence of substratal coupling, cannot be instantiated more than once, oppositely to an *extensive resource*.

–passive resource: subsystem which is not referenced in a function definition at the current instant, but which could eventually be.

–specialized resource: ***1.*** resource which can realize only a particular range objectives ***2.*** *gst.* resource which contains particular parts of the system which are necessary for realizing its objective, without which it cannot operate.

resource dependence: *gst.* constraints to the free evolution of the resource's quantities due to couplings with other elements of the system.

–abstract dependence: coupling with abstract elements of the system.

–substratal dependence: coupling with the physical substrate of the system.

–restriction:

retrospective: *gst.* characteristic of the outputs of a system being functions of the inputs.

S

scope:

self-organization:

sense:

sensor: element of the >*sensory system*.

sensory system: ***1.*** *eng.* set of sensors and associated signal conditioning/fusion functions and resources ***2.*** *gst.* coupling of the perceptor with the perceptive environment, formed by independent quantities.

SOAR: *cognitive architecture* for developing systems which exhibit intelligent behaviour; http:// sitemaker.umich.edu/soar

state: ***1.*** *gst. eng.* a particular set of values of all quantities of the system which may not exist currently, but which is conceived in order to define a particular scenario or situation ***2.*** *gst. eng.* the set of all instantaneous values of all quantities of the system (external and internal) at a particular instant of time [Kli69, p.280] .

–internal state: *gst.* instantaneous contents of the memory which exerts on the functional integrator [Kli69, p.280].

–state-transition structure: *abrv.* ST-structure >*program*.

static knowledge: *gst. referring to a node* part of the knowledge of the system which is referenced (used) for the functional definition of the node >*dynamic knowledge*.

stream: ***1.*** *gst. from the functional point of view, conceptual* minimal framework for processes within a general system, formed by interface, function definition and executor. ***2.*** *gst. from the cognitive point of view* structure for a cognitive component.
–interface: *gst.* specification of interaction channels, procedures and protocols of the function adapted to the rest of the node (and derivated from this, to the rest of the functional structure).
–function definition: *gst.* specification of an algorithm and associated resources.
–executor: *gst.* system resources which perform the function definition.

structural interdependence:
–hypothetic structural interdependence:
–real structural interdependence:

structure: *of a system gst.* constant part of the >*organization* of a system, formed by UC-structure and the ST-structure.
–real structure: *gst.* cause of permanent behaviour.
–hypothetic structure: cause of relatively permanent behaviour.

structure of universe of discourse and couplings (UC-structure): *gst.* [Kli69, p.46] set of couplings and behaviours of the system (part of the >*structure* of the system), real couplings and behaviours form the real structure; hypothetic couplings and behaviours form the hypothetic structure.

substratal coupling: ***1.*** *gst.* coupling between elements of a level of abstraction with the elements of a lower level ***2.*** *gst.* coupling between the abstract and the physical elements of a system.

substratal dependence:

substrate:

Subsumption: *cognitive architecture* based on stimulus-response principle; paradigm of reactive architectures. http://people.csail.mit.edu/brooks/

subsystem: ***1.*** a part of a system which exhibits unified, integrated operation or functionality with respect to the rest of the system ***2.*** module ***3.*** *gst.* a subset of the quantities of a system which is studied separately from the rest, as if it were a system on its own.

success: realization of an objective.

system/object: ***1.*** certain part of nature to which study is confined ***2.*** *gst. when observing a system* set of quantities which are measured according to a space-time specification *usually referring also to the knowledge associated:* time invariant relations, elements, couplings, behaviours, organization, etc.
–closed system: ***1.*** [Kli69, p.70] system in which its principal quantities are not independent, and therefore a there exists a determined boundary with its environment ***2.*** [vB69, p.39] systems which are considered to be isolated from their environment *eg: as in classical physics*.
–controlled system: *gst.* system whose >*control—3* is known.
–general system: ***1.*** system conceived out of particular implementational or substratal aspects, by its functional, structural and operational characteristics ***2.*** *gst.* an abstract model of a particular class of systems [Kli69, p.93].
–open system: ***1.*** system in which the boundary with the environment is undetermined (ie: there exist independent principal quanitites) ***2.*** opposite from >*closed system*.
–autonomous system: ***1.*** system which can operate without human intervention ***2.*** system which can generate and achieve its own objectives under an uncertain environment ***3.*** system which can maintain internal cohesion under the interaction with the environment ***4.*** *gst.* system with high degree

of interdependence and functional capacity >*autonomy*.

–***natural system:*** *gst.* all system which is not ideally-autonomous, *ie:* having at least one independent quantity.

–***totally autonomous system:***

system environment: *gst. in perception* environment of a system.

system time: >*time*.

system time differential: *gst.* minimum period of time in the system resolution level.

T

time:
–***absolute time:***
–***perceptive time:***
–***system time:*** *gst.* sequence of instants of time given by the time resolution level of the system.

time-invariant relation: *gst.* [Kli69, p.39] relation between quantities that is satisfied during a certain time interval.
–***absolute time-invariant relation:*** *gst.* relation which is proper to the given quantities at the given resolution level and which is satisfied over the entire time interval of every particular activity containing the quantities at the resolution level.
–***relative time-invariant relation:*** *gst.* relation which is satisfied during a particular activity containing the quantities at a resolution level.
–***local time-invariant relation:*** *gst.* which applies to shorter time intervals of a particular activity.

time specification: >*resolution level–space-time resolution level*

U

umwelt: *original from Jacob von Uexkull; as in* [vB69, p.227] *biol.* "(. . .) from the great cake of reality, every living organism cuts a slice, which it can perceive and to which it can react owing to its psycho-physical organization." *also referred to as:* reachable/perceivable environment.

universe: ***1.*** [Gra96] all existing matter and space considered as a whole ***2.*** all: material or inmaterial ***3.*** system containing all other systems.

universe of discourse: *gst. of a system* set of all elements of the system. (>*element*.)

V

value: *referring to the measurement of system quantities* the general form of the result of the measurement of a quantity. Some measurements take the form of magnitudes.

variable: *gst.* a quantity without any particular interpretation and dimensionless [Kli69, p.93].

Z

zero coupling: *of a system* relations existing between system elements, $a_1 \dots a_n$ and the remainding element, the environment, a_0. At quantity level, the zero coupling stands for all independent quantities in the system.
–***minimal zero coupling:*** weakest form of substratal coupling in which the zero coupling would be formed by a unique quantity affecting only behaviour of order 0.

Chapter 17

Glossary of Specialized Terms

A

anticipation:

asymmetry: [Ley92, p.7] the memory that processes leave on objects (>symmetry).

ATLANTIS: Cognitive architecture.

auditory adaptation: *phenomenon* the loudness of a continuous tone appears to decrease over time due to repetition or sustained exposure [Sch01, p.356].

auditory fatigue: temporary loss of sensitivity to sounds following the exposure to intense sounds. It may affect differently to the frequencies of the spectrum, analogously to the phenomenon of >masking, the main difference being that auditory fatigue occurs *after* exposure [Sch01, p.356]. It is measured by the >temporary threshold shift (TTS).

autonomy:

awareness:

B

BB1/AIS: Cognitive architecture. Principal researcher: Barbara Hayes-Roth.

beat: subjective perception of two sustained tones with slightly different frequencies. As the difference in frequency grows, the two sounds begin to be perceived separately, starting by a hard and dissonant sensation [Sch01, p.353].

blackboard:

C

characteristic:
–chroma: brightness or dullness of a hue.

circadian rhythms: biological cycles synchronous with day, that regulate the activity of some organs.

consciousness:

consonant sound: pleasant combination of two tones. Subjective characteristic dependent on multiple factors such as culture, custom, learning and other [Sch01, p.358]. >dissonant

constructivism: *in perception* influence of the perceiver on its own process of perception [Hug01, p.8]. *analogously in other senses: constraints put by the actor to its own action*

cue: a piece of information or particular configuration interpreted as a signal or hint by the perceptive system, towards the recognition of an object.

–binocular cues: depth cues resulting of using both eyes simultaneously [WBS92, p.305].
–depth cues: assist in perception of size, depth and distance of figures [WBS92, p.305].
–monocular cues: depth cues resulting from using only one eye, used to create the illusion of three-dimensional space on a two-dimensional plane [WBS92, p.305].

D

density: *of sound* subjective quality, apparently reciprocally related with volume and directly with sound intensity, picturing a certain *compactness* of a tone [Sch01, p.358].

description: [Mar82, p. 20] the result of using a representation to describe a given entity.

difference threshold ΔI: Minimum perceivable difference in sound intensity [Sch01, p.348].

dissonant sound: unpleasant combination of two tones. Subjective characteristic dependent on multiple factors such as culture, custom, learning and other [Sch01, p.358]. >consonant

dominance: condition that occurs when one or more compositional elements within a visual field is emphasized and becomes more visually prominent than the others [WBS92, p.373]. *can be generalized to all senses*.

E

ecological perception: >perception

electroreception: [Hug01, p.201] sensory modality based on electricity, which is present in some marine creatures, which is used to detect proximity of creatures and may be used as a sense of direction (compass).
–active electroreception: said to be possessed by animals that analyze the disturbances introduced by other animals in a self-generated electric field.
–passive electroreception: said to be possessed by animals which can sense only electrical fields generated by other animals.

element: *of a system* smaller part of a system: subsystem. Usually serves to construct the explanation of time-invariant relations by composition relations between elements (>explanation, time-invariant relation). An element is defined by a distinct set of quantities and a time-invariant relation between these quantities at a given resolution level. (>universe of discourse.)

emotion:
–background emotions:
–primary/universal emotions:
–secondary/social emotions:

F

feature: *esp. in artificial vision* characteristic property of a certain object to be recognized, whose appearance and intensity is evaluated in an image or sensor input >invariant.
–feature vector: *esp. in artificial vision* vector in which each coordinate represents the value measured for each feature considered >feature space.

feature space: *esp. in artificial vision* (1) space of the same dimension as the feature vector. (2) in a space of the same dimension as the feature vector, the region in which feature vectors are considered to represent a particular object to be recognized ie.: feature vectors out of this region indicate the image (or sensory input in general) does not contain the object. The boundary of the region is usually called *decision boundary*.

feeling:

figure:

–ambiguous/fluctuating figure: figure with more than one dominant shape, which causes the viewer to visually switch back and forth from one shape to another [WBS92, p.335], for example, the Necker Cube.
–illusionary figure: mental representation of a figure not completely represented, but induced in the viewer by misleading visual clues.
–impossible figure: "a figure which converts two sets of contradictory clues, causing the brain to make conflicting spatial interpretations." [WBS92, p.335].

frame: context or point of view.
–retinocentric frame: *when describing, for example, relations between seen characteristics* regarding to the situation in the viewer's retina.
–viewer-centered frame: from the spatial point of view of the viewer.

function:

functional space:
–atomic function:

G

goodness: *referring to a form or shape,* the degree up to which it participates of certain qualities such as (according to Gestalt): simplicity, regularity, symmetry, and ease of being remembered [WBS92, p.337].

Gestalt: movement in psychology which began in Germany in the 1920s and lasted for approximately 25 years. It was focused on visual perception, memory and association, thinking and learning. These topics were related with the main interest of the movement, the explanation of how and why configurations of form are understood differently when observed isolatedly or in a context. The main assumption is that context influences perception so that it is not equivalent to perceive elements separatedly than in accumulated form. Representative pshychologists of Gestalt were: Max Wertheimer, Wolfgang Köhler and Kurt Koffka.

H

homeostasis: [Gra96] the tendency toward a relatively stable equilibrium between interdependent elements esp. as maintained by physiological processes.

hue: a specific colour or light wavelength found in the spectrum [WBS92, p.239].
–primary hues: hues that cannot be obtained by mixing.

I

image: *inspired from [Mar82]:* a particular sort of representation consisting in an array of pixels.

interface:

invariant: [Gib66] in [Mar82, p.27] permanent property of the environment.

J

:

K

:

L

Law of Closure: visual continuity.

level of description: *when analyzing a process:* [Mar82, p.22-27] there are three levels from abstract to particular: *computational theory*, *representation and algorithm*, and *hardware implementation*.

loudness: subjective or psychological equivalent for sound intensity. Relations between both are complex [Sch01, p.348].

–equal-loudness contours: **isophonic contours**, **Flechter-Munson curves** representation of sounds which are perceived with equal loudness in a graph with intensity of sound against frequency. Each level of loudness produces a curve in the intensity/frequency plane. Loudness is measured in *phones*, a unit referred to decibels.

M

masking: *auditory phenomenon* rise in the threshold of one tone (test tone) due to the presence of a second (masker tone) [Sch01, p.354].
–interaural masking: masking of the tones heard by one ear due to a masker tone perceived with the other.
–line-busy hypothesis: physiological explanation of masking, by which the phenomenon is due to the masker tone exciting the parts of the sensory apparatus sensing the rest of frequencies, thus preventing them from normal activity [Sch01, p.355].

meaning:

memory:

mental model:

N

:

O

:

P

parallax: apparent displacement of a figure as seen from two different points [WBS92, p.305].

perceive: (1) *conventional* to distinguish or observe through the senses: to see, feel, hear, taste or smell [WBS92, p.81].
–direct perception: ecological perception
–ecological perception: Approach to the study of perception initiated by J.J. Gibson. Two main ideas summarize the underlying philosophy: (1) internal mental processes do not affect perception (2) all stimuli for perception are in the environment [Sch01, p.8].
–modules of perception: [Mar82, p.10] specialized parts of perception which can be studied separately.

pitch: subjective, psychological equivalent to frequency [Sch01, p.351].

pixel: (1) picture element, *inspired from [Mar82]:* value of intensity generated by a detector. (2) Atomic component of an image.

point of view:: *from which a system is analyzed.* Defined framework of space-time resolutions, considered quantities and time-invariant relations in terms of which a system is analyzed.

Pragnanz Law: basic law that governs the decomposition of the perceptual field into separate image areas that are easier to identify. It states that "(...) shapes are perceived in as good form as possible" [WBS92, p.337].

proportion: a comparison between size and quantity, usually expressed in ratios [WBS92, p.195].

Q

:

R

:

S

saccade: rapid movement of the eye between fixation points [Gra96].
–akinesia: failure to generate saccades.
–hypometric saccade: undershooting the target.

semantics:

symmetry: (1) the correspondence between opposite halves of a figure on either side of an axis or set of axes [WBS92, p.373]. (2) [Ley92, p.7] the absence of process-memory. Leyton's theory holds that asymetry in figures and shapes comes as a result of transforming an originally symmetrical shape, ie: as a process overcome to an initially *good* shape (>good, Gestalt). Hence, a symmetrical shape is the absence of a transformation or the absence of its track >asymmetry.

T

temporary threshold shift (TTS): increment in the intensity threshold value beyond which a tone of a certain frequency is perceived. Used to quantify >auditory fatigue. The duration of TTS depends on the characteristics of the exposure, and can derivate into a permanent threshold shift (PTS), a form of hearing loss [Sch01, p.356].

U

:

V

vision: >[Mar82, p.31] Vision is a process that produces from images of the external world a description that is useful to the viewer and not cluttered with irrelevant information.

visual field: Region of the external world sensed by the visual system.

volume: *of sound* apparent size, expansiveness, voluminousness of a tone: subjective sound experience complementary to loudness, pitch and density.

W

:

X

:

Y

:

Z

:

Bibliography

[Alb91] James S. Albus. Outline for a theory of intelligence. *IEEE Transactions on Systems, Man and Cybernetics*, 21(3):473–509, May/June 1991.

[Alb95] James S. Albus. RCS: A Reference Model Architecture for Intelligent Systems, 1995. In Working Notes: AAAI 1995 Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents.

[Alb99] James S. Albus. The Engineering of Mind. *Information Sciences*, 117(1-2):1–18, 1999. http://www.isd.mel.nist.gov/documents/albus/engineeringmind-96.pdf.

[Ame99] Scientific American, editor. *The Book of the Brain*. The Lyons Press, 1999.

[Anc99] François Anceau. *Vers une étude objective de la conscience*. Germes Science, 1999.

[And03a] Michael L. Anderson. Embodied cognition: A field guide. *Artificial Intelligence*, 149:91–130, 2003.

[And03b] Michael L. Anderson. Representations, symbols, and embodiment. *Artificial Intelligence*, 149:151–156, 2003.

[Arb72] Michael A. Arbib. *The Metaphorical Brain*. Wiley Interscience, 1972.

[Ark98] Ronald C. Arkin. *Behavior-Based Robotics*. MIT Press, 1998.

[ARMA+94] I. Alarcon, P. Rodriguez-Marin, L.B. Almeida, R. Sanz, L. Fontaine, P. Gomez, X. Alaman, P. Nordin, H. Bejder, and E. de Pablo. Heterogeneous integration architecture for intelligent control systems. *IEEE Intelligent Systems Engineering*, 3(3):138–152, 1994.

[ASM+06] Jaime Alarcón, Rubén Salvador, Félix Moreno, Pedro Cobos, and Ignacio López. A new real-time hardware architecture for road line tracking using a particle filter. In *Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society, IECON'06, Paris, France*, pages 736–741. IEEE Industrial Electronics Society, November 2006.

[Baa97] Bernard J. Baars. *In the Theater of Consciousness*. Oxford University Press, 1997.

[BAB+95] Frank Balbach, Jörn Altmann, Bernd Bieker, Joao Carreira, Diamantino Costa, Geert Deconinck, Alan Grigg, Axel Hein, Johan Vounckx, and Gunda Wenkebach. On the realisation of a fault tolerance concept for massively parallel systems. Technical Report 2.3.9, Institute for Computer Science, IMMD III, Friedrich-Alexander University, Martensstrasse 3, D-91058 Erlangen, 1995.

[Bar99] Lawrence W. Barsalou. Perceptual Symbol Systems. *Behavioral and Brain Sciences*, 22:577–660, 1999.

[Bat01] Gregory Bateson. *Mind and Nature: A Necessary Unity*. Hampton Press, 2001.

[BB06] Aaron S. Benjamin and Randy D. Bird. Metacognitive control of the spacing of study repetitions. *Journal of Memory and Language*, 55:126–137, 2006.

[BBC+03] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi. *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, chapter Shape-based pedestrian detection and localization, pages 328– 333. IEEE, 2003.

[BD05] Martin Borchert and Antje Düsterhöft. *Natural Language Processing and Knowledge Engineering, 2005. Proceedings of IEEE NLP-KE '05.*, chapter Emotions in Speech - Experiments with Prosody and Quality Features in Speech for Use in Categorical and Dimensional Emotion Recognition Environments, pages Emotions in Speech – Experiments with Prosody and Quality Features in Speech for Use in Categorical and Dimensional Emotion Recognition Environments. IEEE, 2005.

[Ber06] Julia Bermejo Alonso. Integrating cognition, emotion and autonomy: A state of the art. Internal document. Autonomous Systems Laboratory (ASLab), May 2006.

[BL03] Glen E. Bodner and D. Stephen Lindsay. Remembering and knowing in context. *Journal of Memory and Language*, 48:563–580, 2003.

[BLMLS05] H. Barth, K. La Mont, J. Lipton, and E. Spelke. Abstract number and arithmetic in preschool children. In 102, editor, *Proceedings of the National Academy of Sciences*, volume 39, pages 14117–14121, 2005.

[BML+06] Hilary Barth, Kristen La Mont, Jennifer Lipton, Stanislas Dehane, Nancy Kanwisher, and Elisabeth Spelke. Non-symbolic arithmetic in adults and young children. *Cognition*, 98:199–222, 2006.

[BOAJ06] Nils Bertschinger, Eckehard Olbrich, Nihat Ay, and Jürgen Jost. Autonomy: An information-theoretic perspective. Working Paper 06-10-035, Santa Fe Institute, 2006.

[Boj04] David M. Boje. Handout on alternative systems theories. Web publication: http://cbae.nmsu.edu/~dboje/655/655_overview.htm, August 2004. Historical perspective on the different branches of the Theory of General Systems.

[Bro91a] Rodney A. Brooks. Integrated Systems Based on Behaviors. *SIGART Bulletin*, 2(4):46–50, 1991.

[Bro91b] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991. people.csail.mit.edu/brooks/papers/representation.pdf.

[Bro99] Alberto Broggi. The argo autonomous vehicle's vision and control systems. *International Journal of Intelligent Control and Systems*, 3(4):409–441, 1999.

[Bry01] Joanna Bryson. *Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents*. PhD thesis, Massachusetts Institute of Technology, September 2001. www.cs.bath.ac.uk/ jjb/ftp/phd.pdf.

[BS01] Joanna Bryson and Lynn Andrea Stein. Modularity and Design in Reactive Intelligence. In *IJCAI*, pages 1115–1120, 2001.

[BSZ06] Nicolas Bredeche, Zhongzhi Shi, and Jean-Daniel Zucker. Perceptual learning and abstraction in machine learning: An application to autonomous robotics. *IEEE Transactions on Systems, Man and Cybernetics*, 36(2):172–181, March 2006.

[BU06] Sara L. Bengtsson and Fredrik Ullén. Dissociation between melodic and rhythmic processing during piano performance from musical scores. *NeuroImage*, 30:272–284, 2006.

[BW97] Alan Burns and Andy Wellings. *Real-Time Systems and Programming Languages*. Addison-Wesley, 1997.

[Can39] Walter B. Cannon. *The Wisdom of the Body*. W. W. Norton, 1939.

[CH00] W. D. Christensen and C. A. Hooker. Autonomy and the emergence of intelligence: Organised interactive construction. *Communication and Cognition - Artificial Intelligence*, 17(3-4):133–157, 2000.

[Cha04] Harold Henry Chaput. *The Constructivist Learning Architecture: A Model of Cognitive Development for Robust Autonomous Robots*. PhD thesis, Computer Science Department, University of Texas at Austin, August 2004. Technical Report TR-04-34.

[Che00] Ben M. Chen. *Robust and H_∞ Control*. Springer, 2000.

[Chr03] Ron Chrisley. Embodied artificial intelligence. *Artificial Intelligence*, 149:131–150, 2003.

[Col] John Collier. Functionality and autonomy in open dynamical systems. http://www.ukzn.ac.za/undphil/collier/papers/carleton.pdf.

[Coo14] Theodore Andrea Cook. *The Curves of Life*. Dover Publications, 1979, first published in 1914.

[Cra] Tim Crane. The problem of perception, the stanford encyclopedia of philosophy (winter 2006 edition), edward n. zalta (ed.).

[Cri93] Flaviu Cristian. Understanding fault-tolerant distributed systems. Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093-0114, May 1993.

[CW04] Lola Cañamero and WP7 members. Preliminary plans for exemplars: Cognition & action (humaine d7b). Public Report (Deliverable) D7b, Human-Machine Interaction Network on Emotions (HUMAINE) IST FP6 507422, May 2004.

[Dam00] Antonio Damasio. *The Feeling of What Happens: Body, Emotion and the Making of Consciousness*. Vintage, 2000.

[Dav96] Paul Davidsson. *Autonomous Agents and the Concept of Concepts*. PhD thesis, Department of Computer Science, Lund University, 1996. http://www.cs.lth.se/publications/ai/1996/PhD_Davidsson.pdf.

[Dav97] E. R. Davies. *Machine Vision*. Academic Press, 2nd edition, 1997.

[DBK91] Mark Drummond, John Bresina, and Smadar Kedar. The entropy reduction engine: integrating planning, scheduling, and control. *ACM SIGART Bulletin*, 2(4):61–65, August 1991. http://delivery.acm.org/10.1145/130000/122355/p61-drummond.pdf.

[Dee01] John Deely. Umwelt. *Semiotica*, 134(1), 2001.

[Den91] Daniel Dennett. *Consciousness Explained*. Penguin, 1991.

[DG00] Subrata Das and Dan Grecu. Cogent: cognitive agent to amplify human perception and cognition. In *Proceedings of the fourth international conference on Autonomous agents*, pages 443–450, 2000.

[DhS01] Richard O. Duda, Perter E. hart, and David G. Stork. *Pattern Classification*. Wiley Interscience, 2001.

[DVC+94] Geert Deconinck, Johan Vounckx, R. Cuyvers, R. Lauwereins, B. Bieker, H. Willeke, E. Maehle, A. Hein, Frank Balbach, Jörn Altmann, M. Dal Cin, H. Madeira, J.G. Silva, R. Wagner, and G. Viehöver. Fault tolerance in massively parallel systems. *Transputer Communications*, 2(4):241–257, December 1994.

[Ell97] Willis D. Ellis. *A Source Book of Gestalt Psychology*. Gestalt Journal Press, 1997.

[Emm01] Claus Emmeche. Does a robot have an umwelt? reflections on the qualitative biosemiotics of jakob von uexküll. *Semiotica*, 134(1):653–693, 2001.

[Far00] Martha J. Farah. *The Cognitive Neuroscience of Vision*. Blackwell, 2000.

[Fer05] Elisabetta Ferrando. *Abductive Processes in Conjecturing and Proving*. PhD thesis, Faculty of Purdue University, May 2005.

[Fir50] Roderick Firth. Sense data and the percept theory. *Mind*, 58 and 59, 1949 and 1950.

[Fis01] Hans Rudi Fischer. Abductive reasoning as a way of worldmaking. *Foundations of Science*, 6(4: special issue on "The Impact of Radical Constructivism on Science", edited by A. Riegler):361–383, 2001.

[Fit97] Paul Fitzpatrick. *A novel behaviour-based robot architecture and its application to building an autonomous robot sentry*. Masters thesis, College of Informatics and Electronics University of Limerick, Ireland, 1997. http://people.csail.mit.edu/paulfitz/pub/paulfitz-masters.pdf.

[FKID03] Masahiro Fujita, Yoshihiro Kuroki, Tatsuzo Ishida, and Toshi T. Doi. Autonomous behavior control architecture of entertainment humanoid robot sdr-4x. In *Proceedings of the 2003 Conference on Intelligent Robots and Systems*, pages 960–967. IEEE, October 2003. http://ieeexplore.ieee.org/iel5/8832/27983/01250752.pdf.

[Flo03] Razvan V. Florian. Autonomous artificial agents. Technical Report Coneural-03-01, Center for Cognitive and Neural Studies (Coneural), Str. Saturn 24, 3400 Cluj-Napoca, Romania, 2003.

[FMD05] Enrique G. Fernández-Abascal, María Dolores Martín Díaz, and Javier Domínguez Sánchez. *Procesos Psicológicos*. Pirámide, 2005.

[Fra95] Stan Franklin. *Artificial Minds*. MIT Press, 1995.

[Fuk75] Kunhiko Fukushima. Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, 20:121–136, 1975.

[Fuk80] Kunhiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[FW95] Walter Fedderwitz and Thies Wittig. Real-time expert systems for industrial control. *Integrated Computer-Aided Engineering*, 2(3):187–202, 1995.

[Gai78] Brian R. Gaines. Progress in general systems research. In *Applied General Systems Research, George J. Klir (ed)*, pages 3–28. Plenum Press, 1978.

[Gar00] Isaac Garrido. *Psicología de la Emoción*. Síntesis Psicología, 2000.

[Gat92] Erann Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots. In *Proc. National Conference on Artificial Intelligence. San Jose, CA*, pages 809–815, 1992. **http://www.flownet.com/gat/papers/aaai92.pdf**.

[Gat98] Erann Gat. On three-layer architectures. In *Artificial Intelligence and Mobile Robots, David Kortenkamp and R. Peter Bonnasso, and Robin Murphy (Eds.)*. AAAI Press, 1998.

[Gat99] Erann Gat. Non-linear sequencing. In *Proceedings of the IEEE Aerospace Conference, 1999*, 1999. http://www.flownet.com/gat/papers/nls.pdf.

[Ghy83] Matila C. Ghyka. *Estética de las Proporciones en la Naturaleza y en las Artes*. Poseidon, 1983.

[Gib65] James J. Gibson. The comparison of mediated perception with direct perception. Purple Perils, Unpuplished manuscripts, web edition, 1965.

[Gib66] James J. Gibson. *The Senses Considered as Perceptual Systems*. Houghton Mifflin, 1966.

[Gib87] James J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1987.

[GL00] Torkel Glad and Lennart Ljung. *Control Theory: Multivariable and Nonlinear Methods*. Taylor & Francis, 2000.

[GL04] Jeremi Gancet and Simon Lacroix. Embedding heterogeneous levels of decisional autonomy in multi-robot systems. *7th International Symposium on Distributed Autonomous Robotic Systems (DARS'04), Toulouse (France) 23-25 Juin*, 2004. Rapport LAAS No04383 http://www.laas.fr/ simon/publis/GANCET-DARS-2004.pdf.

[GL05] Vittorio Gallese and George Lakoff. The brain's concepts: The role of the sensory-motor system in conceptual knowledge. *COGNITIVE NEUROPSYCHOLOGY*, 21(0), 2005.

[GMP$^+$01] Veysel Gazi, Mathew L. Moore, Kevin M. Passino, William P. Shakleford, Frederick M. Proctor, and James S. Albus. *The RCS Handbook: Tools for Real-Time Control Systems Software Development*. John Wiley & Sons, 2001.

[Gog97] Walter C. Gogel. The organization of perceived space. In *Indirect Perception, Irvin Rock (Ed)*, pages 361–383. MIT Press, 1997.

[Gra96] Gramercy Books, editor. *Webster's Encyclopedic Unabridged Dictionary*. Gramercy Books, 1996.

[GW92] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[Har90] S. Harnad. The symbol grounding problem. *Physica*, 42:335–346, 1990.

[Has03] W. Haselager. Form, function and the matter of experience. *SEED*, 3(3):100–111, 2003.

[HBB+06] H. Van Dyke Parunak, Robert Bisson, Sven Brueckner, Robert Matthews, and John Sauter. A model of emotions for situated agents. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. International Conference on Autonomous Agents. Hakodate, Japan.*, pages 993–995, 2006.

[Hey90] Francis Heylighen. Autonomy and cognition as the maintenance and processing of distinctions. In *Self-Steering and Cognition in Complex Systems, F. Heylighen and E. Rosseel and F. Demeyere (Eds)*, pages 89–106. Gordon and Breach, 1990.

[HMH04] Ed. Hui-Min Huang. Autonomy levels for unmanned systems (alfus) framework. volume i: Terminology. version 1.1. Technical Report NIST Special Publication 1011, National Institute for Standards and Technology (NIST), September 2004.

[Hoc74] Julian Hochberg. *Perception: Essays in Honor of J. J. Gibson*, chapter Higher-order stimuli and inter-response coupling in the perception of the visual world, pages 17–39. Cornell University Press, 1974.

[Hol03] Owen Holland, editor. *Machine Consciousness*. Inprint Academic, 2003.

[HR95] Barbara Hayes-Roth. An architecture for adaptive intelligent systems. *Artificial Intelligence*, 72(1-2):329–365, 1995. **http://citeseer.nj.nec.com/hayes-roth95architecture.html**.

[Hue] Michael Huemer. Sense-data, the stanford encyclopedia of philosophy (winter 2006 edition), edward n. zalta (ed.).

[Hug01] Howard C. Hughes. *Sensory Exotica: A World Beyond Human Experience*. MIT Press, 2001.

[Hur01] Susan Hurley. Perception and action: Alternative views. *Synthese*, 129:3–40, 2001.

[Jal94] Pankaj Jalote. *Fault Tolerance in Distributed Systems*. P T R Prentice Hall, 1994.

[KD95] Christoph Koch and Joel L. Davies, editors. *Large-Scale Neuronal Theories of the Brain*. MIT Press, 1995.

[Ken03] Catriona M. Kennedy. *Distributed Reflective Architectures for Anomaly Detection and Autonomous Recovery*. PhD thesis, School of Computer Science. University of Birmingham, June 2003. http://www.cs.bham.ac.uk/research/cogaff/kennedy-phd-thesis.pdf.

[KGESss] Elizabeth A. Kensinger, Rachel J. Garoff-Eaton, and Daniel L. Schacter. Effects of emotion on memory specificity: Memory trade-offs elicited by negative visually arousing stimuli. *Journal of Memory and Language*, 2006 In press.

[Kli69] George J. Klir. *An Approach to General Systems Theory*. Van Nostrand Reinhold, 1969.

[Kli80] George J. Klir. *Teoría de los Sistemas Generales*. Matemática Actual. ICE Ediciones, 1980.

[Kli01] George J. Klir. *Facets of Systems Science*, volume 15 of *IFSR International Series on Systems Science and Engineering*. Kluwer Academic Publishers, 2 edition, 2001.

[Kof63] Kurt Koffka. *Principles of Gestalt Psychology*. Harbinger, 1963.

[Köh59] Wolfgang Köhler. *Gestalt Psychology*. Signet, 1959.

[Köh69] Wolfgang Köhler. *The Task of Gestalt Psychology*. Princeton Univ Press, 1969.

[Kös00] Hatice Köse. *Towards a Robust Cognitive Architecture for Autonomous Robots*. PhD thesis, Bogaziçi University, 2000. http://eliza.cmpe.boun.edu.tr/thesis/haticemsthesis.pdf.

[KS05] Stefan Koelsch and Walter A. Siebel. Towards a neural basis of music perception. *Trends in Cognitive Sciences*, 9(12):578–584, December 2005.

[KSJ00] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessel, editors. *Principles of Neural Science*. McGraw-Hill, 4th edition edition, 2000.

[LBCC99] John E. Laird, Clare Bates Congdon, and Karen J. Coulter. *The Soar User's Manual Version 8.2*, 1 edition, 1999.

[Lev00] Michael W. Levine. *Fundamentals of Sensation and Perception*. Oxford University Press, 3rd edition, 2000.

[Ley92] Michael Leyton. *Symmetry, Causality, Mind*. MIT Press, 1992.

[Lot02] Mihhail Lotman. Umwelt and semiosphere. *Sign Systems Studies*, 30(1), 2002.

[LS97] F. William Lawvere and Stephen H. Schanuel. *Conceptual Mathematics: A first introduction to categories*. Cambridge University Press, 1997.

[Lyo95] William Lyons, editor. *Modern Philosoophy of Mind*. Everyman, 1995.

[MA05] Lilia Moshkina and Ronald C. Arkin. Human perspective on affective robotic behavior: A longitudinal study. In *Proceedings of the International Conference on Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ*, pages 1444–1451. IEEE, August 2005. http://ieeexplore.ieee.org/iel5/10375/32977/01545343.pdf.

[MACJ$^+$95] F. Matía, J. A. Aguilar-Crespo, A. Jiménez, R. Sanz, and J. M. Domínguez. Fuzzy logic and data quality in real-time systems. *Integrated Computer-Aided Engineering*, 2(3):229–239, 1995.

[MAHP03] Félix Moreno, Francisco Aparicio, Wílmar Hernández, and Javier Páez. A low–cost real–time fpga solution for driver drowsiness detection. In *Proceedings of the 29th Annual Conference, IEEE Industrial Electronics Society, IECON'03. Virginia (USA)*. IEEE Industrial Electronics Society, 2003.

[Man00] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman and Company, 19th printing edition, 2000.

[Man04] Jean Matter Mandler. *The Foundations of Mind: Origins of Conceptual Thought*. Oxford Series in Cognitive Development. Oxford University Press, 2004.

[Mar82] David Marr. *Vision*. W. H. Freeman and Company, 1982.

[Mat05] Margaret W. Matlin. *Cognition*. John Wiley & Sons, 6th edition, 2005.

[May86] Richard E. Mayer. *Pensamiento, resolución de problemas y cognición*. Paidós, 1st edition, 1986. Original title: Thinking, Problem Solving, Cognition. Ed. W. H. Freeman & Co.

[McD95] Skot McDonald. Manufacturing industrial: The artificial perception and generation of rhythm. Technical report, Dpt. of Computer Science. University of Western Australia, 1995.

[Mey00] Alexander Meystel. Measuring performance of systems with autonomy: Metrics for intelligence of constructed systems, 2000. White Paper for the Workshop on Performance Metrics for Intelligent Systems. NIST, Gaithesburg, Maryland, August 14-16, 2000.

[MR86] James L. McClelland and David E. Rumelhart. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume II: Psychological and Biological Models. MIT Press, 1986.

[New90] Allen Newell. *Unified Theories of Cognition*. Harvard University Press, 1990.

[Noë04] Alva Noë. *action in perception*. MIT Press, 2004.

[Oga90] Katsuhiko Ogata. *Modern Control Engineering*. Prentice-Hall, 1990.

[OJSK05] O. Oullier, K. Jantzen, F. Steinberg, and J. Kelso. Neural substrates of real and imagined sensorimotor coordination. *Cerebral Cortex*, 15(7):975–985, 2005.

[Ore04] Anders Orebäck. *A Component Framework for Autonomous Mobile Robots*. PhD thesis, Royal Institute of Technology (KTH). Stockholm. Sweden., 2004.

[Pei58] Charles Sanders Peirce. *Collected Papers of Charles Sanders Peirce*. Volumes I-VI edited by Charles Hartshorne and Paul Weiss. Cambridge, Massachusetts, 1931-1935. Volumes VII-VIII edited by Arthur Burks, Cambridge, Massachusetts, 1958, 1931-1935; 1958.

[Per86] Kurt E. Peray. *Rotary Cement Kiln*. CHS Press, 2nd edition, 1986.

[Pic98] Rosalind Picard. *Affective Computing*. MIT Press, 1998.

[Pie91] Benjamin C. Pierce. *Basic Category Theory for Computer Scientists*. Foundations of Computing Series. MIT Press, 1991.

[PJ01] Caroline Palmer and Melissa K. Jungers. Episodic memory for musical prosody. *Journal of Memory and Language*, 45:526–545, 2001.

[POMS05] David Pereira, Eugénio Oliveira, Nelma Moreira, and Luís Sarmento. Towards an architecture for emotional bdi agents. Technical Report DCC-2005-09, Department of Computer Science. University of Porto (Portugal)., 2005.

[Pyl99] Zenon Pylyshyn. Is vision continuous with cognition? the case for cognitive impenetrability of visual perception. *Behavioral and Brain Sciences*, 22:341–364, 1999.

[RC06] Gabriel A. Radvansky and David E. Copeland. Memory retrieval and interference: Working memory issues. *Journal of Memory and Language*, 55:33–46, 2006.

[RK03] Bruno H. Repp and Günther Knoblich. Perceiving action identity: How pianists recognize their own performances. *Psychological Science*, 15(9):604–609, 2003.

[RLN93] Paul S. Rosenbloom, John E. Laird, and Allen Newell, editors. *The Soar Papers*, volume 2. MIT Press, 1993.

[RM86] David E. Rumelhart and James L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume I: Foundations. MIT Press, 1986.

[RN95] Stuart J. Russell and Peter Norvig. *Artificial Intelligence*. Prentice Hall International, 1995.

[Roc83] Irvin Rock. *The Logic of Perception*. MIT Press, 1983.

[Roc85] Irvin Rock. *La Percepción*. Prensa Científica, 1985.

[Roc97] Irvin Rock. *Indirect Perception*. MIT Press, 1997.

[Roy05] Deb Roy. Grounding words in perception and action: computational insights. *Trends in Cognitive Sciences*, 9(8):389–396, August 2005.

[San90] Ricardo Sanz. *Arquitectura de Control Inteligente de Procesos*. PhD thesis, Universidad Politécnica de Madrid, 1990.

[Sch95] David G. Schwartz. *Cooperating Heterogeneous Systems*. Kluwer Academic Publishers, 1995.

[Sch01] Harvey Richard Schifmann. *Sensation and Perception*. John Wiley & Sons, 2001.

[Sel59] O. G. Selfridge. *The Mechanization of Thought Processes*, chapter Pandemonium: A paradigm for learning., pages 511–529. London: H. M. Stationary Office, 1959.

[SGJ+94] R. Sanz, R. Galan, A. Jimenez, F. Matia, J.R. Velasco, and G. Martinez. Computational intelligence in process control. In *IEEE World Congress on Computational Intelligence*, pages 2203–2208, Orlando, FL, USA, 1994.

[SGQ05] Jacqueline Spinola de Freitas, Ricardo R. Gudwin, and João Queiroz. Emotion in artificial intelligence and artificial life research: Facing problems. In *Intelligent Virtual Agents: 5th International Working Conference, IVA 2005*. Springer, 2005.

[Sha] Alexei Sharov. Pragmatism and umwelt-theory. http://www.ento.vt.edu/ sharov/biosem/txt/umwelt.html.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

[Sha05] Murray Shanahan. Perception as abduction: turning, sensor data into meaningful representation. *Cognitive Science*, 29:103–134, 2005.

[Sim90] Herbert A. Simon. *The Sciences of the Artificial*. MIT Press, 1990.

[SJG+92] R. Sanz, A. Jimenez, R. Galan, F. Matia, and E. A. Puente. Intelligent process control: the CONEX architecture. In *Engineering systems with intelligence: concepts, tools and applications, by S. G., Tzafestas (Editor)*, pages 407–413. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[SJS03] W. Carter Smith, Scott P. Johnson, and Elisabeth Spelke. Motion and edge sensitivity in perception of object unity. *Cognitive Psychology*, 46:31–64, 2003.

[SL91] Jean-Jaques Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice-Hall, 1991.

[SMG00] Ricardo Sanz, Fernando Matía, and Santos Galán. Fridges, elephants, and the meaning of autonomy and intelligence. In *Proceedings of the 15th IEEE lntemational Symposium on Intelligent Control (ISIC 2000)*, pages 217–222. IEEE, July 2000. http://ieeexplore.ieee.org/iel5/7088/19098/00882926.pdf.

[SP05] Melissa A. Schilling and Christopher Paparone. Modularity: An application of general systems theory to military force development. *Defense Acquisition Review Journal*, pages 279–293, 2005.

[SS04] Kristin Shutts and Elisabeth Spelke. Straddling the perception-conception boundary. *Developmental Science*, 7(5), 2004.

[SSdA+99] Ricardo Sanz, Miguel J. Segarra, Angel de Antonio, Fernando Matía, Agustín Jiménez, and Ramón Galán. Design patterns in intelligent control systems. In *Proceedings of IFAC 14th World Congress*, Beijing, China, 1999.

[SSK06] Timothy A. Salthouse, Karen L. Siedlecki, and Laci E. Krueger. An individual differences analysis of memory control. *Journal of Memory and Language*, 55:102–125, 2006.

[Sta98] William Stallings. *Operating Systems: Internals and Design Principles*. Prentice-Hall, 3rd edition, 1998.

[Sto] Hans-Georg Stork. Cognition and (artificial) cognitive systems.

[Sut90] Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In Morgan Kaufmann, editor, *Proceedings of the Seventh Conference on Machine Learning*, pages 216–224, 1990. www.cs.ualberta.ca/ bulitko/F02/papers/sutton-90.pdf.

[SZ01] Tarja Susi and Tom Ziemke. Social cognition, artefacts, and stigmergy: A comparative analysis of theoretical frameworks for the understanding of artefact-mediated collaborative activity. *Cognitive Systems Research*, 2(4):273–290, 2001.

[Tay99] John G. Taylor. *The Race for Consciousness*. MIT Press, 1999.

[Tho61] D'Arcy Wentworth Thompson. *On Growth and Form*. Cambridge University Press, 1961.

[Ull80] Shimon Ullman. Against direct perception. *Behavioral and Brain Sciences*, 3:373–381, 1980.

[Ull96] Shimon Ullman. *High-Level Vision*. MIT Press, 1996.

[Uni05] University of Hertfordshire, Hatfield, UK. *Proceedings of the Symposium on Next Generation Approaches to Machine Consciousness: Imagination, Development, Intersubjectivity and Embodiment. AISB'05 Convention: Social Intelligence and Interaction in Animals, Robots and Agents*. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, 12-15 April 2005.

[vB69] Ludvig von Bertalanffy. *General System Theory*. George Braziller, 1969.

[VDL+94] Johan Vounckx, Geert Deconinck, R. Lauwereins, G. Viehöver, R. Wagner, H. Madeira, J.G. Silva, Jörn Altmann, B. Bieker, and H. Willeke. The ftmps-project: Design and implementation of fault–tolerance techniques for massively parallel systems. In *Proceedings of the nternational Conference and Exhibition on High-Performance Computing and Networking*. Springer, 1994.

[vH05] Hermann von Helmholtz. *Treatise on Physiological Optics*, volume III. Dover Phoenix, 1962 edition, 2005.

[VKWL06] Bradley W. Vines, Carol L. Krumhansl, Marcelo M. Wanderley, and Daniel J. Levitin. Cross-modal interactions in the perception of musical performance. *Cognition*, 101:80–113, 2006.

[vU82] Jacob von Uexkull. The theory of meaning. *Semiotica*, 42(1):25–82, 1982.

[WB94] Michael Wolverton and Lee Brownston. *BB1 v3.2 Manual*, 1994.

[WBS92] Charles Wallschlaeger and Cynthia Busic-Snyder. *Basic Visual Concepts and Principles*. McGraw-Hill, 1992.

[WD04] Michael Wertheimer and D. Brett King. *Max Wertheimer and Gestalt Theory*. Transaction Publishers, 2004.

[Web] Principia Cybernetica Web. Cybernetics and systems thinkers. Principia Cybernetica Web http://pespmc1.vub.ac.be/CSTHINK.html.

[Wie06] Jiri Wiedermann. Hugo: A cognitive architecture with an incorporated world model. Technical Report 966, Institute of Computer Science. Academy of Sciences of the Czech Republic., Pod Vodarenskou vezi 2, 182 07 Prague 8, 2006.

[Win67] Fritz Winckel. *Music, Sound and Sensation*. Dover Publications, 1967.

[WJ94] Mark Wooldridge and Nick Jennings. Agent theories, architectures, and languages: A survey. In *Proceedings of ECAI94 Workshop on Agent Theories Architectures and Languages*, pages 1–32, 1994. http://160.97.47.93/eiter/intagnt/jenn-woold-94.pdf.

[WK01] Robert A. Wilson and Frank C. Keil, editors. *The MIT Encyclopedia of Cognitive Sciences*. MIT Press, 2001.

[WLC00] S. Wallace, J. Laird, and K. Coulter. Examining the resource requirements of artificial intelligence architectures. In *Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation*, pages 73–82, May 2000.

[WWPP06] Yingxu Wang, Wing Wang, Shushma Patel, and Dilip Patel. A layered reference model of the brain (lrmb). *IEEE Transactions on Systems, Man and Cybernetics*, 36(2):124–133, March 2006.

[Zad65] Lofti A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[Zad02] Lofti A. Zadeh. From computing with numbers to computing with words - from manipulation of measurements to manipulation of perceptions. *International Journal of Applied Mathematics and Computer Science*, 12(3):307–324, 2002.

[ZJ00] Wayne Zachary and Jean-Christophe Le Mentec. Incorporating metacognitive capabilities in synthetic cognition. In *Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation*, pages 513–521, 2000.

[ZLFK05] T. Zanto, E. Large, A. Fuchs, and J. Kelso. Gamma band responses to auditory sequences: Evidence for synchronization of perceptual processes. *Music Perception*, 22:535–552, 2005.

Index