Chapter 2
Intelligent Control System Architectures

Making realistic robots is going to polarize the market, if you will. You will have some people who love it and some people who will really be disturbed.

David Hanson

Abstract The design of sociorobots can be performed efficiently by exploiting some kind of structured framework, in order to integrate and implement the underlying perception, cognition, learning, control, and social interaction functions. This necessity has motivated the development of many different intelligent control architectures with particular features, advantages, and weaknesses. This chapter starts by providing a discussion of the basic functional design requirements, and an outline of the two early seminal behavior-based control architectures, namely the subsumption and motor schemas architectures. Then, the chapter describes three important newer architectures, namely a 4-layer architecture, the deliberative-reactive architecture, and the combined symbolic/subsumption/servo-control (SSS) architecture. A general discussion and categorization of the characteristics of the intelligent control architectures is also included. All these architectures were used successfully in many available sociorobots.

2.1 Introduction

Human beings have strong interest in others’ labor rather than their own. They used animals and then they invented machines. Today, much effort and money is given for making ‘Intelligent’ machine and beings. The research and development in humanoid robots is part of these efforts, and among others (practical utility of them, etc) offers a good research tool for understanding the human brain and body (cognition, kinematics, dynamics, locomotion, and control). Much of the efforts in anthropomorphic and zoomorphic robots have been and are still made in legged locomotion, arm control, dexterous manipulation, human-robot interaction, learning and adaptive behavior, perception, social performance, etc. Humanoids and animaloids will surely change the way humans interact with machines. Industrial-oriented humanoid robots would considerably increase industrial efficiency and take humans where they have never been.
The design process of sociorobots is inherently inefficient, and so some kind of structured framework is needed to enable profitable integration and implementation of the underlying control, cognition, learning, and social interaction functions. This necessity has motivated the development of a variety of intelligent control architectures with different complexities, features and capabilities. In general, the basic steps for the design and development of any architecture are the following:

- Complex task definition.
- Decomposition of the complex task in simple (generic, primitive) tasks.
- Task solving components development and implementation.
- Integration components’ development and implementation.
- Overall system integration and validation.

The objective of this chapter is:

- To discuss the basic functional requirements for the sociorobot design (high-level/low level cognition, motivation, attention, behavior, motion).
- To outline the two early and seminal behavior-based control architectures (subsumption, motor schemas).
- To describe three newer hierarchical architectures (four-layer, deliberative-reactive, SSS: Symbolic, Subsumption, Servo-control architecture).
- To provide a general discussion and categorization of the characteristics of the available architectures.

### 2.2 Requirements for Sociorobot Design

Sociorobots are robots which, in addition to conventional motion and behavior capabilities, should have the capability to learn and interact directly and bilaterally with human partners. Learning should not be limited to mathematical and numerical model learning, but also to learning human functions including both locomotion and social behaviors. This type of learning and interaction is achieved using the available sensors for vision, speech, position/velocity, tactile/force, etc. In broad terms the general capabilities that a sociorobot must have were listed in Sect. 1.5.2 and include cognition, perception, learning, planning and control. Of particular importance are the cognition and perception capabilities.

*Cognition* is the human mental process of knowing through perception, reasoning or intuition. Etymologically, the word cognition comes from the Greek word ‘γνώση/γνωρίζω’ (gnosi/gnorizo = knowledge/know) and the Latin word ‘cogno-sere’ (to know). The branch of psychology that studies the mental processes that include how people think, perceive, learn, remember, and solve problems is referred to as ‘cognitive psychology’. It is focused on how people acquire, process, and store information.
Perception is the human process or action of perceiving or the product of effect of perceiving (i.e., the insight or intuition gaining via perceiving). Etymologically, comes from the Latin word perception (comprehension). In general, perception is the process by which a living organism detects and interprets information from the environment via sensory receptors.

For the purposes of humanoid sociorobot design the perception process can be divided in high-level and low-level perception [1].

**High-level perception** This includes (non-exhaustively) the following capabilities:
- Gesture recognition (pointing, gaze direction, etc).
- Attention state recognition.
- Social versus non-social object discrimination.
- Recognition of self and other.
- Face and eye detection.
- Speech recognition (sound stream, prosody, phoneme extraction).

**Low-level perception** Here the following capabilities are included:
- Auditory feature extraction
- Visual feature extraction (color, motion, edge detection)
- Tactile and kinesthetic sensing.

The gesture recognition is needed for enabling the robot to perform shared attention functions (e.g., learn from an instructor by attending the same objects and understanding where new information should be used).

In addition to high-and-low-level perception systems, a sociorobot need a ‘motivation system’ composed by two subsystems. The first of them enables the robot to acquire social inputs and understand human cues and emotional states. The second subsystem enables the robot to handle its environment (e.g., to warn the human teacher that he/she is performing too quickly, by frustration). The motivation system is supplemented by an ‘attention system’ which involves mechanisms for habituation, integration of low-level motivation effects. The behaviors of the robot can be coherently incorporated in an overall ‘behavior system’. A typical behavior system includes, but it is not restricted to, the following features: shared and directing attention, arbitration of competing behaviors, vocalization generation, selection, avoidance or orientation of behaviors, etc. Finally, the control and actuation mechanisms for achieving body posture, expressive skills, visual-based skills and manipulation skills (reaching, grasping), etc, are collected in the ‘motor/actuator’ system. These systems, namely: high-level perception system, low-level perception system, motivation system, attention system, behavior system, and motor/actuation system must be interlinked through proper software and hardware interfaces. A possible interlinking scheme is the 3-layered architecture shown in Fig. 2.1.
The low-level percepts are sent to the attention system, which picks-out those which are relevant at that time, and direct the robot’s attention and gaze toward them. The motivation system communicates bilaterally with the attention system, and involves the robot’s basic emotions’ and ‘drives’ (i.e., the basic needs of the robot modeled as simple ‘homeostatic’ regulation mechanisms). The motor system receives commands from the motivation and behavior systems regarding the blending and sequencing the elementary actions from the corresponding specialized motor/actuation devices. The architecture of Fig. 2.1 was basically used for building the MIT sociorobots Cog and Kismet [2, 3].

2.3 Early Generic Behavior-Based Architectures: Subsumption and Motor Schemas Architectures

The two early very popular behavior-based architectures are the ‘subsumption architecture’ developed by Brooks [4], and the ‘motor schemas architecture’ developed by Arkin [5, 6].
2.3.1 Subsumption Architecture

The subsumption architecture which is based on the *sense-plan-act* paradigm (Fig. 2.2) was firstly employed in the autonomous robot *Shakey* [7]. The tasks by which each behavior is achieved are represented as separate layers (Fig. 2.2b) in contrast to the conventional sense-plan-act model (Fig. 2.2a). Individual layers work on individual goals concurrently and asynchronously. At the lowest level the system behavior is represented by an *augmented finite state machine* (AFSM) shown in Fig. 2.3.

**Fig. 2.2** Distinction between the classical sense-plan-act model (a), and the subsumption model (b)

**Fig. 2.3** AFSM employed in the subsumption architecture
The term ‘subsumption’ originates from the verb ‘to subsume’ which means to think about an object as taking part of a group. In the context of behavioral robotics, the term subsumption comes from the coordination process used, between the layered behaviors within the architecture. Complex actions subsume simple behaviors. Each AFSM performs an action and is responsible for its own perception of the world [4, 5]. The reactions are organized in a hierarchy of levels where each level corresponds to a set of possible behaviors. Under the influence of an internal or external stimulation, a particular behavior is required. Then, it emits an influx towards the inferior level. At this level, another behavior arises as a result of simultaneous action of the influx and other stimuli. The process continues until terminal behaviors are activated. A priority hierarchy fixes the topology. The lower levels in the architecture have no awareness of higher levels. This allows the use of incremental design. That is, higher-level competencies are added on top of an already working control system without any modification of those lower levels.

2.3.2 Motor Schemas Architecture

The motor schemas architecture was more strongly motivated by biological sciences and uses the theory of schemas, the origin of which goes back to the 18th century (Immanuel Kant). Schemas represent a means by which understanding is able to categorize sensory perception in the process of realizing knowledge of experience. The first applications of schema theory include an effort to explain postural control mechanisms in humans, a mechanism for expressing models of memory and learning, a cognitive model of interaction between motor behaviors in the form of schemas interlocking with perception in the context of the perceptual cycle, and a means for cooperation and competition between behaviors.

From among the various definitions of the schema concept available in the literature we give here the following representative ones [6, 8]:

- A pattern of action or a pattern for action.
- An adaptive controller which is based on an identification procedure for updating the representation of the object under control.
- A perceptual entity corresponding to a mental entity.
- A functional unit that receives special information, anticipates a possible perceptual content, and matches itself to the perceived information.

A convenient working definition is the following [8]: ‘A schema is the fundamental entity of behavior from which complex actions can be constructed, and which consists of the knowledge how to act or perceive, as well as the computational process by which it is enacted’.

Using schemas, robot behavior can be encoded at a coarser granularity than neural networks while maintaining the features of concurrent cooperative-competitive
control involved in neuroscientific models. More specifically, schema theory-based analysis and design of behavior-based systems possesses the following capabilities:

- It can explain motor behavior in terms of the concurrent control of several different activities.
- It can store both how to react and how to realize this reaction.
- It can be used as a distributed model of computation.
- It provides a language for connecting action and perception.
- It provides a learning approach via schema elicitation and schema tuning.
- It can explain the intelligence functions of robotic systems.

Motor schema behaviors are relatively large grain abstractions, which can be used in a wide class of cases. Typically, these behaviors have internal parameters which offer extra flexibility in their use. Associated with each motor schema there is an embedded perceptual schema which gives the world specific for that particular behavior and is capable of providing suitable stimuli.

Three ways in which planning (deliberative) and reactive behavior can be merged are [8]:

- Hierarchical integration of planning and reaction (Fig. 2.4a)
- Planning to guide reaction, i.e. permitting planning to select and set parameters for the reactive control (Fig. 2.4b)
- Coupled planning-reacting, where these two concurrent activities, each guides the other (Fig. 2.4c).

![Fig. 2.4 a Hierarchical hybrid deliberative-reactive structure, b Planning to guide reaction scheme, c Coupled planning and reacting scheme](image-url)
One of the first robotic control schemes that were designed using the hybrid deliberative (hierarchical) and reactive (schema-based) architecture is the Autonomous Robot Architecture (AuRA) [9]. AuRA incorporated a traditional planner that could reason over a modular and flexible behavior-based control system.

### 2.4 A Four-Layer Sociorobot Control Architecture

Here, a four-layer general-purpose sociorobot control architecture (framework) will be described where the layers of abstraction are clear and help in systematically integrating the various tasks and actions of the robot, such as the information-processing task, human tracking, gesture recognition, prediction of human behavior, dynamic path planning, etc [10].

The four layers of the architecture in bottom-up order are the following (Fig. 2.5):

- **Robot driver layer** This is the lower layer which involves all hardware-specific driver modules for sensing (e.g., camera, microphone, laser ranger finder: LRF), and actuation (motor drivers, speaker, etc). This layer allows the use of the same

![The 4-layer sociorobot control architecture](Fig. 2.5)
applications and behaviors to different similar robots (e.g., legged humanoid robots) with minor differences in their size or joint configurations.

- **Information processing layer** This layer includes *sensing* modules (related to localization, human tracking, face detection, speech recognition, sound source localization, etc) and *actuation* modules (performing tasks like gaze following, path following, etc). The nonverbal behaviors are distinguished in *implicit* (not needing to be specified by the designer), and *explicit* (needing to be specified by utterances) [11].

- **Behavior layer** This layer combines sensor processing and actuation behaviors and is designed according to the subsumption behavior-based concept. Representative behavior modules include speech, gesture, timing, and approach modules. Behaviors are implemented as software agents that react to sensor inputs and execute actions including social actions. Behavior modules can also be configured by the designer from the application layer thus enabling the development of flexible, reusable behavior modules [4, 5].

- **Application Layer** At this highest layer, designers can develop social robot applications using an *interaction composer* (a graphical interaction development environment). The graphical representation of the interaction composer (IC) provides a bridge between designers and programmers through their direct mapping to the underlying software modules. The interaction composer enables the designer to configure behaviors in several ways without the need to know the details of the program embedded in the behavior.

Details and experimental results are provided in [10]. This generic interaction design control architecture enables designers and programmers to work in parallel for developing a variety of applications for sociorobots.

### 2.5 A Deliberative-Reactive Control Architecture

This architecture provides an efficient ego-centric robot control environment for the robot cooperation with humans, other robots, and virtual avatars. It is actually a methodology exploiting a combination of *Belief-Desire-Intention* (BDI) agents, a reactive behavioral system, and an explicit social infrastructure. The architecture involves four interlinked sections (levels) as shown in Fig. 2.6 [12]:

- Physical level
- Reactive (behavioral) level
- Deliberative level
- Social level

**Physical level** This level is designed such that the architecture can be used with different robot platforms (including sensors, digital signal processors, motor
controllers, and motors). It can be individually tailored to each hardware platform from wheeled robots to legged humanoid robots.

**Reactive level** This level supervises the physical level by a set of *primitive modules* (behaviors and activities). *Activities* are charged with the tasks of sensors data acquisition and data processing such as feature extraction, etc. Behavioral modules implement the reflex robot responses to real events (unexpected or dangerous), constituting the primary survival components (skills) of the robot. The body of the robot is controlled at any given time by a unique behavior from the available ones. Since the behavior implementations do not refer to the specifics of

![Hybrid deliberative-reactive control architecture](image-url)

*Fig. 2.6* Hybrid deliberative-reactive control architecture
what body they are controlling, the transfer of the code from simulation to real robots is very easy.

**Deliberative level** This level is organized as a *multi-agent-system (MAS)* where various agents supervise the different functional levels of the robot. The control of the robotic platform at any specific time is shared by a number of agents which have complexities varying from simple procedural to means-ends reasoning. The reasoning skill is provided by an integrated and tooled environment for the rapid development of social intentional agents based on BDI agent theory. This environment is known as *Agent Factory* [13]. In BDI systems the deliberative layer is modeled using mental processes that correspond to informational, motivational and deliberative states of the agents. As it is typical in BDI systems, the Agent Factory agents use proper reasoning procedures to deliberate upon their percepts, update the mental state, and select the way for future action.

**Social level** The agents that are available in the *Agent Factory*, in addition to been able to reason about themselves, they can also reason about the features of the other agents they meet. The collaboration of sociorobots is achieved using a specific formalism based on *Speech Act Theory*, which provides a precise and expressive communication tool in multi-agent systems. In [12] the interaction of robotic agents is implemented using the *Agent Communication Language (ACL) Teanga* [14]. Full details of these issues are provided in a series of papers published by Duffy and collaborators [12–16].

The reflex behaviors shown in the reactive level of Fig. 2.6 correspond to a library of behaviors for people to play football with a number of robots in free environments (ball following, dribbling, passing, and kicking). On line video demonstrations that show the success and robustness of the architecture are provided at www.cs.usd.ie/csprise [12].

### 2.6 The SSS Hybrid Control Architecture

This architecture involves three layers namely: Servo-control layer, Subsumption layer, and Symbolic layer, and is known with the acronym ‘**SSS architecture**’. It combines in a convenient way the best properties of standard servo-control systems and signal processing systems, with the capabilities of subsumption-based reactive control, and symbolic representations of state-based control schemes (Fig. 2.7) [17, 18].

This architecture is mostly suitable for robot navigation tasks, where simple linear servo-controllers provide non-accurate results due to their inability to work with uncertain or nonlinear systems. On the other hand, behavior-based systems can deal with uncertain and nonlinear systems since they impose weaker constraints. However, they can work at small sampling rates and usually lead to ‘jerky’ motions as in the Shakey robot. This drawback of behavior-based systems is faced through the use of suitable servo controllers that can provide smooth motions. The other drawback of behavior-based controllers seems to be their ‘distributed nature’, which does not allow to find a good place for the representation and description of the
world model, although their developers argue that actually the distributed nature is their greatest advantage. For many robotic applications, hierarchical (centralized) schemes, that can be implemented by standard symbolic programming languages, are more convenient.

Actually, the SSS architecture is compatible with the fact that servo-controllers operate both in continuous time and continuous state space domain (i.e., they continuously measure the world’s state representing it as an ensemble of scalar values). Behavior-based controllers work in continuous time but discretize the possible world states into a small number of special task-dependent categories. Symbolic systems discretize both the time (on the basis of significant events) and state space (discrete-event control). For an effective integration of the above different methodologies (servo-control, behavior-based control, symbolic-based control) special interfaces are needed (see Chap. 3). The first type of interfaces must be able to transform behavior-based signals to the underlying servo signals (e.g., via matched filters). The interface between the symbolic and the subsumption layer is simply an on–off selective switching of each behavior (via a mechanism that looks for the first time instant at which various situation recognizers are valid). The rate of control is increasing as we go from the symbolic level, to the subsumption level and finally to the servo-control level.

Considering the robot navigation problem, the global (strategic) planning is handled by the symbolic layer which has a coarse geometric map of the robot’s environment, and the local navigation (e.g., wall following) is performed at the reactive control layer. The moment-to-moment navigation is performed by the servo-controllers.
2.7 General Discussion of Sociorobot Control Architectures

From the presentation of the limited, but representative, number of architectures discussed in Sects. 2.2–2.6 one can see that different architectures offer different sets of capabilities achieved by using several concepts, techniques and computational tools. Most of them have a multi-layer hierarchical structure, but with different interpretations of the ‘layer/level’ concept. Unfortunately it seems very difficult (if not impossible) to find a ‘unique’ meaning of what a layer/level characterizes and involves. However, it is possible to find a set of properties, capabilities, and modes of operation that may cover the available architectures. One of these sets, very useful in categorizing the architectures, is the following [19–21]:

- Sequential versus concurrent layers
- Increasing precision with decreasing intelligence layers
- Direct versus trainable control
- Learning repertory
- Intrinsic versus derivative motives
- Single versus multi-window perception
- Fixed versus emergent emotions
- External language requirements
- Centralized, decentralized, and distributed systems.

In sequential layer processing the information comes from bottom and get abstracted as it moves through higher intermediate levels to the top. In concurrent processing architectures different layers are all concurrently active (processed).

In increasing precision with decreasing intelligent architectures (that follow the traditional management or military command style) the higher levels completely dominate (control) the lower levels.

In cases where the higher levels do not directly control the lower levels they may train them to perform certain control actions. For example, a deliberative layer may cause a reactive layer to develop new condition-action behavior sequences, which may afterwards run without supervision.

Typically, a robot control architecture involves several kinds of learning mechanisms in different parts of the system, e.g., neural nets, trainable reactive systems, extendable knowledge bases, etc.

Architecture also differ in the perception way. Perception can be performed in one of two ways. (i) Using ‘single-window’ (peephole) models, in which case information is passed through the hierarchy, or (ii) using a ‘multi-window’ model in which case perceptual processing is layered concurrently and produces different levels of resolution (low at the top and high at the bottom of the hierarchy).

Some architectures have box labeled ‘emotions’, whereas in others emotions are emergent properties of interactions between functional components.

Some architectures are designed using a close link between high-level internal processes and an external language (e.g., via meta-management mechanisms).
Other architectures consider internal formalisms and mechanisms for deliberation and high-level self-evaluation as precursors to the development of natural human language.

In centralized systems all decisions are made by a central control mechanism and sent to the executive components. In decentralized systems each executive sub-system makes its own decisions and executes only these decisions. Finally, in distributed systems the decision is made through a negotiation among the executive subsystems and executed by them. All these system types can be described, explained and implemented using the ‘agent’ concept which involves three parts (communicator for connecting the agent to the head of other agents on the same communication level or higher, head for planning and action selection, and body for action execution).

An effort to develop a biologically inspired framework to cover a variety of architectures with different subsets of components is described in [19] making a three-fold division between perception, central processing, and action. This framework (called CogAff architecture schema) does not cover the distributed type architectures.

2.8 Summary

To design an efficient sociorobot system that satisfies our expectations, the designer should use a suitable architecture that systematizes the specification, representation and integration of the underlying desired functions and behaviors. Unavoidably there does not exist a unique, uniform, and globally accepted architecture. This is firstly due to the unavailability of standardized hardware and software platforms in the market, and to the different modeling, information processing, and control ways that a designer can select to achieve a certain goal. Most critically, in sociorobots, this is due to the ambiguity in specifying and understanding human social behaviors that have to be embodied in the robot.

The present chapter has provided an introduction to the sociorobot control system architectures that are available for use in research and development environments. Specifically, the chapter has discussed the requirements for sociorobot design, and five architectures, namely the subsumption, motor schemas, four-layered, hybrid, deliberative-reactive, and the SSS hybrid architecture. A general discussion of the characteristics of the various architectures has also been included. The material of the chapter is purely conceptual in compatibility with the purpose of the book. Systemic hardware and software implementation details can be found in the references of the chapter and other related references contained therein.
References

Sociorobot World
A Guided Tour for All
Tzafestas, S.G.
2016, XII, 223 p. 121 illus., 90 illus. in color., Hardcover
ISBN: 978-3-319-21421-4