Chapter 2
Simplicity

Overview A recent report on *Software for Dependable Systems: Sufficient Evidence?* [Jac07] by the National Academies contains as one of its central recommendations: *One key to achieving dependability at reasonable cost is a serious and sustained commitment to simplicity, including simplicity of critical functions and simplicity in system interactions. This commitment is often the mark of true expertise*. We consider *simplicity* to be the antonym of *cognitive complexity* (in the rest of this book we mean *cognitive complexity* whenever we use the word *complexity*). In every-day life, many embedded systems seem to move in the opposite direction. The ever-increasing demands on the functionality, and the non-functional constraints (such as safety, security, or energy consumption) that must be satisfied by embedded systems lead to a growth in system complexity.

In this chapter we investigate the notion of cognitive complexity and develop guidelines for building *understandable* computer systems. We ask the question: *What does it mean when we say we understand a scenario?* We argue that it is not the embedded system, but the *models* of the embedded system that must be simple and understandable relative to the *background knowledge* of the observer. The models must be based on clear concepts that capture the relevant properties of the scenario under investigation. The *semantic content* of a program variable is one of these concepts that we investigate in some detail. The major challenge of design is the building of an *artifact* that can be modeled at different levels of abstraction by models of adequate simplicity.

This chapter is structured as follows. Section 2.1 focuses on the topic of cognition and problem solving and an elaboration of the two different human problem-solving subsystems, the *intuitive-experiential* subsystem and the *analytic-rational* subsystem. *Concept formation* and the *conceptual landscape*, that is the private knowledge base that a human develops over his lifetime, are discussed in Sect. 2.2. Section 2.3 looks at the essence of model building and investigates what makes a task difficult. Section 2.4 deals with the important topic of emergence in large systems.

H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Real-Time Systems Series, DOI 10.1007/978-1-4419-8237-7_2,

2.1 Cognition

Cognition deals with the study of thought processes and the interpretation and binding of sensory inputs to the existing knowledge base of an individual [Rei10]. It is an interdisciplinary effort that stands between the humanities, i.e., philosophy, language studies, and social science on one side and the natural sciences, such as neural science, logic, and computer science on the other side. The study of model building, problem solving, and knowledge representation forms an important part of the cognitive sciences.

2.1.1 Problem Solving

Humans have two quite different mental subsystems for solving problems: the *intuitive-experiential subsystem* and the *analytic-rational subsystem* [Eps08]. Neuro-imaging studies have shown that these two subsystems are executed in two different regions of the human brain [Ami01]. Table 2.1 compares some of the distinguishing characteristics of these two subsystems.

> **Example:** A typical task for the *intuitive-experiential subsystem* is *face recognition*, a demanding task that a baby at the age of 6 months can accomplish. A typical task for the *analytic-rational subsystem* is the *confirmation* of a *proof of a mathematical theorem*.

The experiential subsystem is a preconscious emotionally-based subsystem that operates holistically, automatically, and rapidly, and demands minimal cognitive resources for its execution. Since it is nearly effortless, it is used most of the time. It is assumed that the experiential subsystem has access to a large coherent

Table 2.1 Intuitive experiential versus analytic rational. Problem solving strategy (Adapted from [Eps08, p. 26])

Intuitive experiential	Analytic rational
Holistic	Analytic
Emotional (what feels good)	Logical reason oriented (what is sensible?)
Unreflective associative connections	Cause and effect connections, causal chains
Outcome oriented	Process oriented
Behavior mediated by vibes from past experience	Behavior mediated by conscious appraisal of events
Encodes reality in concrete images, metaphors and narratives	Encodes reality in abstract symbols, words, and numbers
More rapid processing, immediate action	Slower processing, delayed action
Slow to change the fundamental structure: changes with repetitive or intense experience	Changes more rapidly, changes with the speed of thought
Experience processed passively and pre-consciously, seized by our emotions	Experience processed actively and consciously, in control of our thoughts
Self evidently valid: *seeing is believing*	Requires justification via logic and evidence

knowledge base that represents an implicit model of the world. This subjective knowledge base, which is one part of what we call the *conceptual landscape* of an individual, is mainly built up and maintained by experience and emotional events that are accumulated over the lifetime of an individual. Although this knowledge base is continually adapted and extended, its core structure is rather rigid and cannot be changed easily. *Experiential reasoning* is holistic and has the tendency to use limited information for general and broad classifications of scenarios and subjects (e.g., this is a *good* or *bad* person). The experiential system does assimilate the data about reality in a coherent stable conceptual framework. The concepts in this framework are mostly linked by *unconscious associative connections*, where the *source* of an association is often *unknown*.

The rational subsystem is a conscious analytic subsystem that operates according to the laws of causality and logic. Bunge [Bun08, p. 48] defines a *causality relationship* between a *cause C* and an *event E* as follows: *If C happens, then (and only then) E is always produced by it*. We try to get an understanding of a dynamic scenario by isolating a *primary cause*, suppressing seemingly irrelevant detail, and establishing a *unidirectional causal chain* between this primary cause and an observed *effect*. If cause and effect cannot be cleanly isolated, such as is the case in a feedback scenario, or if the relationship between cause and effect is *non-deterministic* (see also Sect. 5.6.1 on the definition of *determinism*), then it is more difficult to understand a scenario.

> **Example:** Consider the analysis of a car accident that is caused by *the skidding* of a car. There are a number of conditions that must hold for skidding to occur: the speed of the car, the conditions of the road (e.g., icy road), the conditions of the tires, abrupt manoeuver by the driver, the non-optimal functioning of the computer based skid-control system, etc. In order to *simplify* the model of the situation (the reality is not simplified) we often *isolate* a primary cause, e.g., the *speed*, and consider the other conditions as secondary.

The rational subsystem is a verbal and symbolic reasoning system, driven by a controlled and noticeable mental effort to investigate a scenario. Adult humans have a conscious *explicit model* of reality in their rational subsystem, in addition to their *implicit model* of reality in the experiential subsystem. These two models of reality coincide to different degrees and form jointly the *conceptual landscape* of an individual. There seem to be a nearly unlimited set of resources in the experiential subsystem, whereas the cognitive resources that are available to the rational subsystem are limited [Rei10].

There are many subtle interrelationships between these two problem-solving subsystems, which form the extremes of a continuum of problem solving strategies where both systems cooperate to arrive at a solution. It is not infrequent that, after unsuccessful tries by the rational subsystem, at first a solution to a problem is produced unconsciously by the experiential subsystem. Afterwards this solution is justified by analytical and logical arguments that are constructed by the rational subsystem.

Similarly, the significance of a new scenario is often recognized at first by the experiential subsystem. At a later stage it is investigated and analyzed by the rational subsystem and rational problem solving strategies are developed. Repeated

encounters of similar problems – the accumulation of *experience* – effortful learning and drill move the problem-solving process gradually from the rational subsystem to the experiential subsystem, thus freeing the cognitive resources that have previously been allocated to the problem solving process in the limited rational subsystem. There exist many practical examples that demonstrate this phenomenon: learning a foreign language, learning a new sport, or learning how to drive a car. It is characteristic for a *domain expert* that she/he has mastered this transition in her/his domain and mainly operates in the effortless experiential mode, where a fast, holistic and intuitive approach to problem solving dominates.

> **Example:** A brain-imaging study of the chess-playing strategy of amateurs versus grandmasters investigated the activity in different sections of the brain immediately after a chess move by the partner. The amateurs displayed the highest activity in the *medial temporal lobe* of the brain, which is consistent with the interpretation that their mental activity is focused on the rational analysis of the new move. The highly skilled grandmasters showed more activity in the *frontal and parietal cortices,* indicating that they are retrieving stored information about previous games from expert memory in order to develop an *understanding of the scenario* [Ami01].

2.1.2 Definition of a Concept

In a changing world, *knowledge* about permanent and characteristic properties of objects and situations must be identified and maintained since such knowledge is of critical importance for survival. This knowledge is acquired by the process of *abstraction*, by which the particular is subordinated to the general, so that what is known about the general is applicable to many particulars. Abstraction is a fundamental task of the human cognitive system.

> **Example:** Face recognition is an example for the powerful process of *abstraction*. Out of many particular images of the face of a person – varying angles of observation, varying distance, changing lighting conditions – characteristic permanent features of the face are identified and stored in order that they can be used in the future to recognize the face again. This demanding abstraction process is executed unconsciously, seemingly without effort, in the experiential subsystem. Only its results are delivered to the rational subsystem.

Abstraction forms categories, where a *category* is a set of elements that share common *characteristic features*. The notion of category is *recursive*: the *elements of a category* can themselves be *categories*. We thus arrive at a hierarchy of categories, going from the concrete to the abstract. At the lowest level we find immediate sensory experiences.

A *concept* is a category that is augmented by a *set of beliefs* about its relations to other categories [Rei10, pp. 261–300]. The set of beliefs relates a *new concept* to already *existing concepts* and provides for an *implicit theory* (a subjective mental model). As a new domain is penetrated, new concepts are formed and linked to the concepts that are already present in the conceptual landscape. A concept is a mental construct of the *generalizable aspects* of a known entity. It has an intension (*What is*

the essence?) and an extension, answering the question as to which *things* and *mental constructs* are exemplars of the concept. A concept can also be considered as a *unit of thought* [Vig62].

2.1.3 Cognitive Complexity

What do we mean when we say *an observer understands a scenario*? It means that the concepts and relationships that are employed in the representation of the scenario have been adequately linked with the conceptual landscape and the methods of reasoning of the observer. *The tighter the links are, the better is the understanding*. *Understanding* (and therefore *simplicity*) is thus a *relation* between an observer and a scenario, *not* a *property* of the scenario.

We take the view of Edmonds [Edm00] that *complexity* can only be assigned to *models of physical systems*, but not to the physical systems themselves, no matter whether these physical systems are natural or man made. A physical system has a nearly infinite number of properties – every single transistor of a billion-transistor *system-on-chip* consists of a huge number of atoms that are placed at distinct positions in space. We need to abstract, to build *models* that leave out the seemingly irrelevant detail of the micro-level, in order to be able to reason about properties of interest to us at the macro-level.

What then is a good measure for *the cognitive complexity* of a *model*? We are looking for a quantity that measures the cognitive effort needed to understand the model by a human observer. *We consider the elapsed time needed to understand a model by a given observer a reasonable measure for the cognitive effort and thus for the complexity of a model relative to the observer*. We assume that the *given observer* is representative for the intended user group of the model.

According to the *scientific tradition*, it would be desirable to introduce an objective notion of cognitive complexity without reference to the subjective human experience. However, this does not seem to be possible, since cognitive complexity refers to a *relation* between an objective external scenario and the subjective internal conceptual landscape of the observer.

The perceived complexity of a model depends on the relationship between the existing subjective conceptual landscape and the problem solving capability of the observer versus the concepts deployed in the representation of the model, the interrelations among these concepts and the notation used to represent these concepts. If the observer is an expert, such as the chess grandmaster in the previous example, the experiential subsystem provides an understanding of the scenario within a short time and without any real effort. According to our metric, the scenario will be judged as *simple*. An amateur has to go through a tedious cause-and-effect analysis of every move employing the rational subsystem that takes time and explicit cognitive effort. According to the above metric, the same chess scenario will be judged as *complex*.

There are models of behavior and tasks that are *intrinsically difficult to comprehend* under any kind of representation. The right column of Table 2.2 in Sect. 2.5 lists some characteristics of intrinsically difficult tasks. It may take a long time, even for an expert in the field, to gain an understanding of a model that requires the comprehension of the behavior of difficult tasks – if at all possible. According to the introduced metric, these models are classified as exceedingly complex.

In order to gain an understanding of a large system we have to understand many models that describe the system from different viewpoints at different abstraction levels (see also Sect. 2.3.1). The cognitive complexity of a large system depends on the number and complexity of the different models that must be comprehended in order to understand the complete system. The time it takes to understand all these models can be considered as a measure for the *cognitive complexity of a large system*.

Case studies about the understanding of the behavior of large systems have shown that the *perceptually available information* plays an important role for developing an understanding of a system [Hme04]. *Invisible information flows* between *identified subsystems* pose a considerable barrier to understanding.

If every embedded system is one of its kind and no relationships between different instances of systems can be established, then there is hardly a chance that experience-based expert knowledge can be developed and the transition from the tedious and effortful rational subsystem to the effortless experiential subsystem can take place.

One route to simplification is thus the development of a *generic model of an embedded system* that can be successfully deployed in many different domains at a proper level of abstraction. This model should contain few orthogonal mechanisms that are used recursively. The model must support simplification strategies and make public the internal information flow between identified subsystems, such that the process of gaining an understanding of the behavior is supported. By getting intimately acquainted with this model and gaining experience by using this model over and over again, the engineer can incorporate this model in the experiential subsystem and become an expert. It is one stated goal of this book to develop such a generic cross-domain model of embedded systems.

2.1.4 Simplification Strategies

The resources in the rational problem solving subsystem of humans, both in storage and processing capacity, are limited. The seminal work of Miller [Mil56] introduced a limit of five to seven chunks of information that can be stored in short-term memory at a given instant. Processing limitations are established by the *relational complexity theory* of Halford [Hal96]. Relational complexity is considered to correspond to the *arity* (number of arguments) of a relation. For example, binary relations have two arguments as in LARGER-THAN (elephant, mouse). The

relational complexity theory states that the upper limits of adult cognition seem to be relations at the quaternary level.

If a scenario requires cognitive resources that are beyond the given limits, then humans tend to apply simplification strategies to reduce the problem size and complexity in order that the problem can be tackled (possibly well, possibly inadequately) with the limited cognitive resources at hand. We know of four strategies to simplify a complex scenario in order that it can be processed by the limited cognitive capabilities of humans: *abstraction, partitioning*, *isolation*, and *segmentation:*

- *Abstraction* refers to the formation of a higher-level concept that captures the essence of the problem-at-hand and reduces the complexity of the scenario by omitting irrelevant detail that is not needed, given the purpose of the abstraction. Abstraction is applied *recursively*.
- *Partitioning* (also known as *separation of concerns*) refers to the *division* of the problem scenario into nearly independent parts that can be studied successfully in isolation. Partitioning is at the core of *reductionism*, the preferred simplification strategy in the natural sciences over the past 300 years. Partitioning is not always possible. It has its limits when *emergent properties* are at stake.
- *Isolation* refers to the suppression of seemingly irrelevant detail when trying to find a *primary cause*. The primary cause forms the starting point of the causal chain that links a sequence of events between this primary cause and the observed effect. There is a danger that the simplification strategy of *isolation* leads to a too simplistic model of reality (see the example on skidding of a car in Sect. 2.1.1).
- *Segmentation* refers to the *temporal decomposition* of intricate behavior into smaller parts that can be processed *sequentially*, one after the other. Segmentation reduces the amount of information that must be processed in parallel at any particular instant. Segmentation is difficult or impossible if the behavior is formed by highly concurrent processes, depends on many interdependent variables and is strongly non-linear, caused by positive or negative feedback loops.

2.2 The Conceptual Landscape

The notion of *conceptual landscape,* or the *image* [Bou61], refers to the *personal knowledge base* that is built up and maintained by an individual in the experiential and rational subsystem of the mind. The knowledge base in the experiential subsystem is *implicit*, while the knowledge base in the rational subsystem is *explicit*. The conceptual landscape can be thought of as a structured network of interrelated *concepts* that defines the *world model*, the *personality,* and the *intentions* of an individual. It is built up over the lifetime of an individual, starting from pre-wired structures that are established during the development of the *genotype* to the *phenotype*, and continually augmented as the individual interacts with its environment by exchanging messages via the sensory systems.

2.2.1 Concept Formation

The formation of concepts is governed by the following two principles [And01]:

- The *principle of utility* states that a new concept should encompass those properties of a scenario that are of utility in achieving a stated purpose. The purpose is determined by the human desire to fulfill basic or advanced needs.
- The *principle of parsimony* (also called *Occam's razor*) states that out of a set of alternative conceptualizations that are of comparable utility the one that requires the least amount of mental effort is selected.

There seems to be a *natural level of categorization*, neither too specific nor too general, that is used in human communication and thinking about a domain. We call the concepts at this natural level of categorization *basic-level concepts* [Rei01, p. 276].

> **Example:** The basic level concept *temperature* is more fundamental than the sub-concept *oil-temperature* or the encompassing concept *sensor data*.

Studies with children have shown that basic-level concepts are *acquired earlier* than *sub-concepts* or *encompassing* concepts. As a child grows up it continually builds and adds to its *conceptual landscape* by observing regularities in the perceptions and utility in grouping properties of perceptions into new categories [Vig62]. These new categories must be interlinked with the already existing concepts in the child's mind to form a consistent *conceptual landscape*. By abstracting not only over perceptions, but also over already existing concepts, new concepts are formed.

A new concept requires for its formation a number of experiences that have something in common and form the basis for the abstraction. *Concept acquisition* is normally a bottom-up process, where sensory experiences or basic concepts are the starting point. Examples, prototypes and feature specification play an important role in concept formation. A more *abstract concept* is understood best *bottom up* by generalizations from a set of a suitable collection of examples of already acquired concepts. Abstract analysis and concrete interpretation and explanation should be intertwined frequently. If one remains only at a *low-level of abstraction* then the amount of non-essential detail is overwhelming. If one remains only at a *high-level of abstraction*, then relationships to the world as it is experienced are difficult to form.

In the *real world* (in contrast to an *artificial world*), a *precise definition* of a concept is often not possible, since many concepts become fuzzy at their boundaries [Rei10, p. 272].

> **Example:** How do you define the concept of *dog*? What are its characteristic features? Is a dog, which has lost a leg, still a dog?

Understanding a new concept is a matter of establishing connections between the new concept and already familiar concepts that are well embedded in the conceptual landscape.

> **Example:** In order to understand the new concept of *counterfeit money*, one must relate this new concept to the following already familiar concepts: (1) the concept of *money*,

(2) the concept of a *legal system*, (3) the concept of a *national bank* that is *legalized* to print money and (4) the concept of *cheating*. A *counterfeit money bill* looks like an *authentic money bill*. In this situation, *examples* and *prototypes* are of limited utility.

In the course of cognitive development and language acquisition, *words* (*names*) are associated with concepts. The *essence of a concept* associated with a word can be assumed to be the *same* within a natural language community (*denotation*), but different individuals may associate different *shades of meaning* with a concept (*connotation*), dependent on their *individual existing conceptual landscape* and the differing personal emotional experiences in the acquisition of the concept.

Example: If communicating partners refer to different concepts when using a word or if the concept behind a word is not well established in the (scientific) language community, (i.e., does not have a well-defined denotation), then effective communication among partners becomes difficult to impossible.

If we change the language community, the names of concepts will be changed, although the *essence of the concept, its semantic content,* remains the same. The names of concepts are thus relative to the context of discourse, while the *semantic content* remains invariant.

Example: The semantic *content* of the concept *speed* is precisely defined in the realm of physics. Different language communities give different names to the same concept: in German *Geschwindigkeit*, in French *vitesse*, in Spanish *velocidad*.

2.2.2 Scientific Concepts

In the world of science, new concepts are introduced in many publications in order to be able to express new *units of thought*. Often these concepts are named by a *mnemonic*, leading to, what is often called, *scientific jargon*. In order to make an exposition *understandable*, new concepts should be introduced sparingly and with utmost care. A new scientific concept should have the following properties [Kop08]:

- *Utility*. The new concept should serve a useful well-defined purpose.
- *Abstraction and Refinement*. The new concept should abstract from lower-level properties of the scenario under investigation. It should be clear which properties *are not parts* of the concept. In the case of refinement of a *basic-level concept*, it should be clearly stated what additional aspects are considered in the refined concept.
- *Precision*. The characteristic properties of the new concept must be precisely defined.
- *Identity*. The new concept should have a distinct identity and should be significantly different from other concepts in the domain.
- *Stability*. The new concept should be usable uniformly in many different contexts without any qualification or modification.

- *Analogy*. If there is any concept in the existing *conceptual landscape* that is, in some respects, analogous to the new concept, this similarity should be pointed out. The analogy helps to establish links to the *existing conceptual landscape* of a user and facilitates understanding. According to [Hal96, p. 5]:

  > Analogical reasoning mechanisms are important to virtually every area of higher cognition, including language comprehension, reasoning and creativity. Human reasoning appears to be based less on an application of formal laws of logic than on memory retrieval and analogy.

The availability of a useful, well defined, and stable set of concepts and associated terms that are generally accepted and employed by the scientific community is a mark for the maturity of a scientific domain. An ontology is a shared taxonomy that classifies terms in a way useful to a specific application domain in which all participants share similar levels of understanding of the meaning of the terms [Fis06, p. 23]. Progress in a field of science is intimately connected with concept formation and the establishment of a well-defined ontology.

> **Example:** The main contributions of Newton in the field of mechanics are not only in the formulation of the laws that bear his name, but also in the isolation and conceptualization of the abstract notions *power*, *mass*, *acceleration* and *energy* out of an unstructured reality.

Clear concept formation is an essential prerequisite for any formal analysis or formal verification of a given scenario. The mere replacement of fuzzy concepts by formal symbols will not improve the understanding.

2.2.3 The Concept of a Message

We consider a *message* as a *basic concept* in the realm of communication. A message is an *atomic unit* that captures the value domain and the temporal domain of a unidirectional information transport at a level of abstraction that is applicable in many diverse scenarios of human communication [Bou61] and machine communication. A basic message transport service (BMTS) transports a message from a sender to one or a set of receivers. The BMTS can be realized by different means, e.g., biological or electrical.

For example, the message concept can be used to express the information flow from the human sensory system to the conceptual landscape of an individual. The message concept can also model the indirect high-level interactions of a human with his environment that are based on the use of language.

> **Example:** We can model the sensory perception, e.g. of temperature, by saying that a message containing the sensed variable (temperature) is sent to the *conceptual landscape*. A message could also contain verbal information about the temperature at a location that is outside the realm of direct sensory experience.

The message concept is also a *basic concept* in the domain of distributed embedded computer systems at the architecture level. If the BMTS between encapsulated subsystems is based on *unidirectional temporally predictable multicast messages*, then

the data aspect, the timing aspect, the synchronization aspect, and the publication aspect are integrated in a single mechanism. The BMTS can be refined at a lower level of abstraction by explaining the transport mechanism. The transport mechanism could be wired or wireless. The information can be coded by different signals. These refinements are relevant when studying the implementation of the message mechanism at the physical level, but are irrelevant at a level where the only concern is the timely arrival of the information sent by one partner to another partner.

A *protocol* is an abstraction over a sequence of *rule-based* message exchanges between communicating partners. A protocol can provide additional services, such as flow control or error detection. A protocol can be understood by breaking it down to the involved messages without the need to elaborate on the concrete transport mechanisms that are used.

2.2.4 Semantic Content of a Variable

The concept of a *variable*, a fundamental concept in the domain of computing, is of such importance for the rest of the book that it justifies some special elaboration. A variable can be considered as a *language construct* that assigns an *attribute* to a *concept*. If the point in real-time, the *instant,* when this assignment is valid, is of relevance, then we call the *variable* a *state variable*. As time progresses, the attribute of a state variable may change, while the concept remains the same. A variable thus consists of two parts, a *fixed part*, the *variable name* (or the *identifier*), and a *variable part* called the *value of the variable* that is assigned to the variable. The variable name designates the concept that determines *what we are talking about*. In a given context, the variable name – which is analogous to the name of a concept in a natural language community – must be unique and point to the same concept at all communicating partners. The meaning that is conveyed by a variable is called the *semantic content* of the variable. As we will show in the latter part of this section, the semantic content of a variable is *invariant* to a change in representation. The requirement of *semantic precision* demands that the concept that is associated with a variable name and the domain of values of the variable are unambiguously defined in the model of the given application.

> **Example:** Consider the variable name *engine-temperature* that is used in an automotive application. This concept is too abstract to be meaningful to an automotive engineer, since there are different temperatures in an automotive engine: the temperature of the oil, the temperature of the water, or the temperature in the combustion chamber of the engine.

The unambiguous definition of a concept does not only relate to the meaning of the concept associated with the variable, but also to the specification of the *domain of values* of the variable. In many computer languages, the *type of a variable*, which is introduced as an attribute of the variable name, specifies primitive attributes of the value domain of the variable. These primitive attributes, like *integer* or *floating point number*, are often not sufficient to properly describe all relevant attributes of the value domain. An extension of the type system will alleviate the problem.

Example: If we declare the *value domain* of the variable *temperature* to be *floating point* we still have not specified whether the temperature is measured in units of *Celsius*, *Kelvin* or *Fahrenheit*.

Example: The Mars Climate Orbiter crash occurred because the ground-based software used different system units than the flight software. The first of the recommendations in the report of the mishap investigation board was *that the MPL (Mars Polar Lander) project verify the consistent use of units throughout the MPL spacecraft design and operation* [NAS99].

In different language communities, different variable names may be used to point to the same concept. For example, in an English speaking language community *the temperature of the air* may be abbreviated by *t-air*, while a German speaking community may call it *t-luft*. If we change the representation of the value domain of a variable, e.g., if we replace the units for measuring the temperature from *Celsius* to *Fahrenheit* and adapt the value of the variable accordingly, the *semantic content* expressed by the variable remains the same.

Example: On the *surface* the two variables *t-air* = 86 and *t-luft* = 30 are completely different since they have different names and different values. If, however, *t-air* and *t-luft* refer to the same concept, i.e., the temperature of the air, and the value of *t-air* is expressed in degrees Fahrenheit and that of *t-luft* in degrees Celsius, then it becomes evident that the *semantic contents* of these two variables are the same.

These differences in the representations of the *semantic content* of a variable become important when we look at *gateway components* which link two subsystems of a *system of systems* that have been developed by two different organizations according to two different *architectural styles*. The term *architectural style* refers to all *explicit* and *implicit* principles, rules and conventions that are followed by an organization in the design of a system, e.g., the representation of data, protocols, syntax, naming, and semantics, etc.. The gateway component must translate the variable names and representations from one architectural style to the other architectural style, while keeping the semantic content *invariant*.

Data that describe the properties of (*object*) *data* is sometimes called *meta-data*. In our model of a variable, data that describes the properties of the *fixed parts of a variable* is *meta data*, while the *variable part* of a variable, *the value set*, is (*object*) *data*. *Meta data* thus describes the properties of the concept that is referred to by the variable name. Since *meta data* can become *object data* of another level, the distinction between *data* and *meta data* is relative to the viewpoint of the observer.

Example: The price of a product is *data*, while the currency used to denote the price, the time interval and the location where this price is applicable are *meta data*.

2.3 The Essence of Model Building

Given the rather limited cognitive capabilities of the rational subsystem of the human mind we can only develop a rational understanding of the world around us if we build *simple models* of those properties that are of relevance and interest to

us and disregard (abstract from) detail that proves to be irrelevant for the given purpose. *A model is thus a deliberate simplification of reality with the objective of explaining a chosen property of reality that is relevant for a particular purpose*.

> **Example:** The purpose of a model in *Celestial Mechanics* is the explanation of the movements of the heavenly bodies in the universe. For this purpose it makes sense to introduce the abstract concept of a *mass point* and to reduce the whole diversity of the world to a single *mass point in space* in order that the interactions with other mass points (heavenly bodies) can be studied without any distraction by unnecessary detail.

When a new level of abstraction (a new model) is introduced that successfully conceptualizes the properties relevant for the given purpose and disregards the rest, *simplicity* emerges. Such simplicity, made possible by the formation of proper concepts, give rise to new insights that are at the roots of the *laws of nature*. As Popper [Pop68] points out, due to the inherent imperfection of the *abstraction* and *induction* process, laws of nature can only be falsified, but never be proven to be absolutely correct.

2.3.1 Purpose and Viewpoint

At the start of any modeling activity, a clear purpose of the model must be established. Formulating the precise questions the model must address helps to concretize the purpose of the model. If the purpose of a model is not crystal clear, or if there are multiple divergent purposes to satisfy, then it is not possible to develop a *simple* model.

> **Example:** The purpose of a *model of behavior* of a real-time computer system is to provide answers to the question at *what points in real-time* will the computer system produce *what kind of outputs*. If our computer system is a *System-on-Chip* (*SoC*) with a billion transistors, then we must find a hierarchy of behavioral models to meet our purpose.

The recursive application of the *principles of abstraction* leads to such a hierarchy of models that Hayakawa [Hay90] calls the *abstraction ladder*. Starting with *basic-level concepts* that are essential for understanding a domain, more general concepts can be formed by *abstraction* and more concrete concepts can be formed by *refinement*. At the lowest level of the abstraction ladder are the direct sensory experiences.

> **Example:** The *Four Universe Model* of Avizienis [Avi82] introduces a hierarchy of models in order to simplify the description of the behavior of a computer system. At the lowest level of the hierarchy, the *physical level*, the analog signals of the circuits are observed, such as the rise time of the voltages as a transistor performs a switching operation. The analysis of a circuit behavior at the physical (analog) level becomes difficult as soon as more and more transistors get involved (*emerging complexity*). The next higher level, the *digital logic level*, abstracts from the physical analog quantities and the dense time and introduces binary logic values (*high* or *low*) of signals at discrete instants, resulting in a much simpler representation of the behavior of an elementary circuit, e.g. an AND gate (*emerging simplicity*). Complexity creeps in again as we combine more and

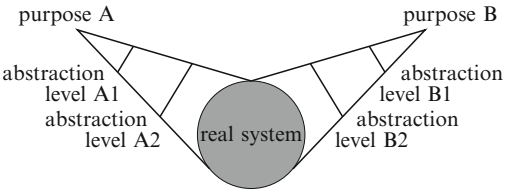

Fig. 2.1 Purpose and abstraction level of a model

more logic circuits. The next higher level, the *information level*, lumps a (possible large) sequence of binary values into a meaningful data structure, (e.g., a pointer, a real-valued variable or a complete picture) and introduces powerful high-level operations on these data structures. Finally, at the *external level,* only the services of the computer system to the environment, as seen by an outside user, are of relevance.

A posed question about a distinct property of a *real system* gives rise to the construction of a hierarchy of models of that system that are intended to answer the posed question. Figure 2.1 depicts two hierarchies of models that are introduced to serve two purposes, purpose A and purpose B. Purpose A could refer to a hierarchy of behavioral models, while purpose B could refer to a hierarchy of dependability models of the same *real system*. At the top of each hierarchy is the stated purpose. i.e., the questions that must be answered. The different levels of the hierarchy – the abstraction levels – are introduced to support a stepwise refinement of the stated question considering more detail, where each step takes consideration of the limited cognitive capabilities of the human mind. At the low end of the hierarchy is the real system. The analysis is substantially simplified if the structure of the model corresponds with the structure of the system. Otherwise we have to resolve a structure clash that complicates the issues.

Example: The model for predicting the temporal properties of the behavior of a real-time computer system is straightforward if there is a predictable sequence of computational and communication actions between the start of a computation and the termination of a computation. Conversely, if the actual durations of the computational and communication actions depend on global system activity (e.g., arbitration for access to shared resources such as caches, communication links, etc.) then it will not be possible to construct a *simple model* for predicting the temporal properties of the behavior.

2.3.2 The Grand Challenge

Whereas the natural scientist must uncover the regularities in a given reality and find appropriate concepts at a suitable level of abstraction in order to formulate models and theories that explain the observed phenomena, the computer scientist is – *at least theoretically* – in a much better position: The computer scientist has the freedom to design the system – *an artifact* – which is the subject of his modeling. The requirement to build artifacts, the properties of which can be analyzed by *simple models*, should thus be an explicit design driver. In many areas of computer science this principle of building artifacts that can be modeled by *simple models* is violated.

For example, the temporal behavior of a modern pipelined microprocessor with multiple caches cannot be captured in a *simple model*.

The major challenge of design is the building of a software/hardware artifact (an embedded computer system) that provides the intended behavior (i.e. the service) under given constraints and where relevant properties of this artifact (e.g., the behavior) can be modeled at different levels of abstraction by models of adequate simplicity.

As stated before, there are many different purposes that give rise to a hierarchy of models of an artifact. Examples are: behavior, reliability, man–machine interaction, energy consumption, physical dimension, cost of manufacturing, or cost of maintenance, to name a few. Out of these, the most important one is the *model of behavior*. In the context of real-time systems, behavior specifies the output actions of a computer system as a consequence of the inputs, the state and the progression of real-time. Output actions and input can be captured in the concepts of *input messages* and *output messages*. In Chap. 4 of this book we present a cross-domain model for the behavior of a real-time computer system using these concepts.

2.4 Emergence

We speak of *emergence* when the interactions of subsystems give rise to unique global properties at the system level that are not present at the level of the subsystems [Mor07]. Non-linear behavior of the subsystems, feedback and feed forward mechanisms, and time delays are of relevance for the appearance of emergent properties. Up to now, the phenomenon of emergence is not fully understood and a topic of intense study.

2.4.1 Irreducibility

Emergent properties are irreducible, holistic, and novel – they disappear when the system is partitioned into its subsystem. Emergent properties can appear unexpectedly or they are planned. In many situations, the first appearance of the emergent properties is unforeseen and unpredictable. Often a fundamental revision of state-of-the-art models is required to get a better understanding of the conditions that lead to the intended emergence. In some cases, the emergent properties can be captured in a new conceptualization (model) at a higher level of abstraction resulting in an *abrupt simplification* of the scenario.

Example: The emergent properties of a *diamond*, such as *brilliance* and *hardness*, which are caused by the *coherent* alignment of the Carbon-atoms, are substantially different from the properties of graphite (which consists of the *same atoms*). We can consider the diamond with its characteristic properties a new concept, a *new unit of thought*, and forget about its composition and internal structure. Simplicity comes out as a result of the intricate interactions among the elements that help to generate a *new whole* with its new emergent properties.

2.4.2 Prior and Derived Properties

When dealing with emergence, it is helpful to distinguish between the *prior properties* of the components and the new *derived properties* that come about by the interactions of the components.

> **Example:** The high reliability of the services of a fault-tolerant system (*derived property*) that is the result of the interactions of many unreliable components (*prior property*) is an emergent property.

In many cases the prior properties and the derived properties can be of a completely different kind. It often happens that the *derived properties* open a completely new domain of science and engineering. This new domain requires the formation of novel concepts that capture essential properties of this new domain.

> **Example:** The property of *being able to fly* which comes about by the proper interaction of the subsystems of an airplane, such as the wings, the fuselage, the engines and the controls, is only present in the airplane as a whole but not in any of the isolated subsystems. *Being able to fly* has opened the domain of the air transportation industry with its own rules and regulations. For example, the subject of *air traffic control* is far removed from the prior properties of the components that make up an airplane.

Prior properties and derived properties are relative to the viewpoint of the observer. When climbing up the abstraction ladder, the derived properties at one level of abstraction become the prior properties at the next higher level of abstraction and so on, since a new form of emergence can appear at higher levels.

> **Example:** In the evolution of the universe two very significant stages of emergence are the *appearance of life* and at a further stage the *appearance of consciousness* that forms the basis for the development of human culture. The realm of human culture has developed its own system of concepts in the arts, sciences etc., that are far removed from the biological prior properties that are characterizing the human brain.

Emergent behavior cannot be predicted analytically, but must be detected in an operating system. Thus control elements must incorporate hooks for monitoring system performance in real time [Par97, p. 7]. The multicast message concept, discussed in Sect. 2.2.3 provides the basis for the nonintrusive observation of system behavior.

2.4.3 Complex Systems

We classify a system as *complex* if we are not in the position to develop a set of models of *adequate simplicity* – commensurate to the rational capabilities of the human mind – to explain the structure and behavior of the system. In addition to *life* and *consciousness*, examples for complex systems are the earth's climate and weather, the global economy, living organisms, and many large computer systems, to name a few.

We hold the opinion that a fundamental understanding of a complex system can only be achieved by a *proper conceptualization* and not by the execution of

elaborate computer simulations. This view is also shared by Mesarovic et al. [Mes04, p.19] when he speaks about biology:

> We further argue that for a deeper understanding in systems biology investigations should go beyond building numerical mathematical or computer models – important as they are .... Such a categorical perspective led us to propose that the core of understanding in systems biology depends on the search for organizing principles rather than solely on construction of predictive descriptions (i.e. models) that exactly outline the evolution of systems in space and time. The search for organizing principles requires an identification/discovery of new concepts and hypotheses.

Maybe, sometimes in the future, we will form appropriate concepts that will lead to an abrupt simplification of some of today's complex systems. If this happens, the system will not be classified as complex any more.

Whereas system biology deals with a natural system, a large computer system is an *artifact* developed by humans. When designing such an artifact, we should take consideration of the limited rational problem solving capability of humans in order that we can describe the behavior of the artifact by models of adequate simplicity. These models should guide the design process, such that a structure clash between the model and the artifact is avoided.

> **Example:** Let us look at the technical example of designing the on-chip communication infrastructure for the communication among IP-cores on a system-on-chip. There are basically two technical alternatives, the provision of a *shared memory* that can be accessed by all IP-cores or the provision of *local memory to each one of the IP-cores* and the design of a message-passing subsystem that enables the exchange of messages among IP-cores [Pol07,Lev08]. The message-passing subsystem isolates and makes explicit the global communication among subsystems and thus supports the introduction of a new level in the hierarchy where a distinction is made between the *intra-IP core* interactions and the *inter-IP core* interactions. The common memory intermixes global intra-IP-core and local inter-IP-core interactions and makes it very difficult to separate global and local concerns, leading to a more complex system model.

2.5 How Can We Achieve Simplicity?

Cognitive scientists have studied how students learn and understand different tasks [Fel04]. They have identified a set of task characteristics that require a disproportional mental effort for understanding the task. Table 2.2 compares the characteristics of *simple* tasks versus *difficult* tasks. We thus need to design a generic model for expressing the behavior of an embedded system that avoids the characteristics of difficult tasks. It should be possible to apply the model *recursively*, such that large systems can be modeled at different levels of abstraction using the same modeling mechanisms.

The model of a real-time system, presented in Chap. 4, tries to reach this goal. Simplicity is achieved by adhering to the following seven design principles:

1. *Principle of Abstraction*. The introduction of a component (a hardware/software unit) as a basic structural and computational unit makes it possible to use

Table 2.2 Characteristics of simple versus difficult tasks (Adapted from [Fel04, p. 91])

Characteristics of a simple task	Characteristics of a difficult task
Static: The properties of the task do not change over time.	Dynamic: The properties of the task are time dependant.
Discrete: The variables that characterize the task can only take values from discrete sets.	Continuous: The domain of the variables is continuous.
Separable: Different subtasks are nearly independent. There is only a weak interaction among tasks.	Non-separable: Different subtasks are highly interactive. It is difficult to isolate the behavior of a single task.
Sequential: Behavior can be understood by a sequential step-by-step analysis.	Simultaneous: Many concurrent processes interact in generating visible behavior. Step-by-step analysis is difficult.
Homogeneous: Components, explanatory schemes, and representations are alike.	Heterogeneous: Many different components, explanatory schemes, and representations.
Mechanism: Cause and effect relations dominate.	Organicism: Behavior characterized by a multitude of feedback mechanisms.
Linear: Functional relationships are linear.	Non-linear: Functional relationships are non-linear.
Universal: Explanatory principles do not depend on context.	Conditional: Explanatory principles are context dependent.
Regular: Domain characterized by a high regularity of principles and rules.	Irregular: Many different context dependent rules.
Surface: Important principles and rules are apparent by looking at observable surface properties.	Deep: Important principles are covert and abstract and not detectable when looking at surface properties.

the component on the basis of its precise interface specifications without any need to understand the internals of the component operation. In order to maintain the abstraction of a component even in the case that faults are occurring, a component should be a fault-containment unit (Sect. 6.1.1). If components stand in a hierarchical relationship to each other, different *levels of abstraction* can be distinguished. At a high level of abstraction, the behavior of a complete autonomous *constituent system* (consisting of many clusters of components) of a system-of-systems (SoS) is captured in the precise linking interface specification of its gateway component (see Sects. 4.6 and 4.7.3).

2. *Principle of Separation of Concerns*. This principle helps to build simple systems by disentangling functions that are separable in order that they can be grouped in self-contained architectural units, thus generating *stable intermediate forms* [Sim81]. This principle is sometimes called *principle of partitioning* [Ses08]. An example is the strict separation of computational activities from communication activities such that the communication system and the computational components can be developed independently (Sect. 4.1.1).
3. *Principle of Causality*. The analytical-rational problem solving subsystem of humans excels in reasoning along causal chains. The *deterministic behavior* of basic mechanisms makes it possible that a causal chain between a cause and the consequent effect can be established without a doubt (Sect. 5.6).

4. *Principle of Segmentation.* This principle suggests that hard-to-understand behavior should be decomposed, wherever possible, into a serial behavioral structure such that a sequential step-by-step analysis of the behavior becomes possible. Each step requires only the investigation of the limited context that is of relevance at this step.
5. *Principle of Independence.* This principle suggests that the interdependence of architectural units (*components* or *clusters*, see Sect. 1.1) should be reduced to the necessary minimum that is required by the application. An example is the provision of a single unidirectional primitive for the communication among components such that any low-level dependency of the sender of a message on the correct operation of the receiver is eliminated by design. This principle is of paramount importance in the design of fault-tolerant systems to ensure that back-propagation of failures is avoided and the independence of failures of fault-containment units can be assumed (Sect. 6.4).
6. *Principle of Observability.* Non-visible communication channels among architectural units pose a severe impediment for the understanding of system behavior. This can be avoided by supporting a multicast topology in the basic message passing primitive. It is then possible to observe the external behavior of any component without a *probe* effect (Sect. 12.2).
7. *Principle of a Consistent Time.* The progression of real-time is an important independent variable in any behavioral model of the physical subsystem of an embedded system. This principle suggests that *a global time base* should be introduced in the distributed computer system such that system-wide consistent temporal relations (e.g., simultaneity) and temporal distances among events can be established on the basis of global time-stamps (Sect. 3.3). The availability of a global time simplifies the solution of many problems in distributed systems (see Sect. 14.2.1).

Points to Remember

- Humans have two quite different mental subsystems for solving problems: the *intuitive-experiential subsystem* and the *analytic-rational subsystem*.
- The *experiential subsystem* is a preconscious emotionally-based subsystem that operates holistically, automatically, and rapidly, and demands minimal cognitive resources for its execution.
- The *rational subsystem* is a conscious analytic subsystem that operates according to the laws of logic. It is well equipped to handle deterministic relations and *causality*.
- Adult humans have a conscious *explicit model* of reality in their rational subsystem, in addition to their *implicit model* of reality in the experiential subsystem. These two models of reality coincide to different degrees and form jointly the *conceptual landscape* of an individual.
- Knowledge is acquired by the process of *abstraction*, by which the particular is subordinated to the general, so that what is known about the general is applicable to many particulars.

- A *concept* is a category that is augmented by a *set of beliefs* about its relations to other categories. The set of beliefs relates a *new concept* to already *existing concepts* and provides for an *implicit theory* (a subjective mental model).
- *Understanding* means that the concepts and relationships that are employed in the representation of a scenario have been adequately linked with the conceptual landscape and the methods of reasoning of the observer. *The tighter the links are, the better is the understanding*. *Understanding* (and therefore *simplicity*) is thus a *relation* between an observer and a scenario, *not* a *property* of the scenario.
- The elapsed time needed to understand a model by an intended observer is a reasonable measure for the cognitive effort and thus for the *complexity* of a *model relative to the observer*.
- *Complexity* can only be assigned to models of *physical systems*, but not to the physical systems themselves, no matter whether these physical systems are natural or man made.
- The complexity of a large system depends on the number and complexity of the models that must be comprehended in order to understand the complete system. The time it takes to understand all these models can be considered as a measure for the *cognitive complexity of a large system*.
- Invisible information flows between *identified subsystems* pose a considerable barrier for understanding.
- The resources in the rational problem solving subsystem of humans, both in storage and processing capacity, are limited.
- The four strategies to simplify a complex scenario in order that it can be processed by the limited cognitive capabilities of humans are *abstraction, partitioning*, *isolation*, and *segmentation*.
- The formation of concepts is governed by the following two principles the *principle of utility* and the *principle of parsimony* (also called *Occam's razor*).
- The *essence of a concept,* i.e., *the semantic content of a concept*, associated with a *name,* can be assumed to be the *same* within a natural language community (*denotation*), but different individuals may associate different *shades of meaning* with a concept (*connotation*).
- A *variable* is a *language construct* that assigns an *attribute* to a *concept* at the given *instant*. A variable thus consists of two parts, a *fixed part*, the *variable name,* and a *variable part* called the *value of the variable* that is assigned to the variable at a particular instant.
- Differences in the representations of the *semantic content* of a variable become important when we look at *gateway components* which link two subsystems that have been developed by two different organizations according to two different *architectural styles*.
- A *model* is a deliberate simplification of reality with the objective of explaining a chosen property of reality that is relevant for a particular purpose.
- If the purpose of a model is not crystal clear, or if there are multiple divergent purposes to satisfy, it is not possible to develop a *simple* model.

- The recursive application of the *principles of abstraction* leads to such a hierarchy of models. More general models can be formed by *abstraction* and more concrete models can be formed by *refinement*.
- The major challenge of design is the building of a software/hardware artifact (an embedded computer system) that provides the intended behavior (i.e. the service) under given constraints and where relevant properties of this artifact (e.g., the behavior) can be modeled at different levels of abstraction by models of adequate simplicity.
- We talk about e*mergence* when the interactions of subsystems give rise to unique global properties at the system level that are not present at the level of the subsystems. Emergent properties are irreducible, holistic, and novel – they disappear when the system is partitioned into its subsystems.
- We classify a system as *complex* if we are not in the position to develop a set of models of *adequate simplicity* – commensurate to the rational capabilities of the human mind – to explain the structure and behavior of the system.

Bibliographic Notes

The textbook by Reisberg [Rei10] gives a good overview of the state-of-the-art in the field of cognition and introduces many of the terms that have been used in this chapter. Epstein [Eps08] discusses the characteristics of the *intuitive-experiential subsystem* and the *analytic-rational subsystem* of problem solving. Boulding [Bou61] elaborates extensively on the notion of *conceptual landscape* (which he calls the *Image*) and the role of the *message metaphor* in all types of communication. The hierarchy of concepts, the *abstraction ladder*, is taken from Hayakawa [Hay90]. The relational complexity theory of Halford [Hal96] establishes limits for the rational reasoning capability of humans, while Miller [Mil56] elaborates on the limits of the human short-term memory capacity. Popper [Pop68] and Edmonds [Edm00] discuss the relevance and limitations of model building for understanding physical systems. The book by Bedau [Bed08] is devoted to the topic of emergence. The comparison of simple versus difficult tasks is taken from [Fel04]. The PhD thesis by Rumpler [Rum08] deals with *design comprehension* of embedded real-time systems. Roger Session's book [Ses08] *Simple Architectures for Complex Enterprises* contains practical guidelines for designing understandable Enterprise Information Architectures.

Review Questions and Problems

2.1 What are the distinguishing characteristics of the *intuitive-experiential* and the *analytic-rational* subsystems for human problem solving?

2.2 Give concrete examples for typical tasks for the *intuitive-experiential* and the *analytic-rational* problem solving subsystems!

2.3 How is a *concept* defined? What are the principles that guide *concept formation*? What is the *conceptual landscape*? What are basic level concepts?
2.4 What is characteristic for a *domain expert*?
2.5 What do we mean when we say we *understand a scenario*? How is *cognitive complexity* defined? Give an example of a *barrier to understanding*!
2.6 Which are known *simplification strategies*?
2.7 What are the characteristics of scientific concepts?
2.8 What is the concept of a message? What is a *protocol?*
2.9 What is the *semantic content* of a variable? What is the relationship between *the representation of a variable* and its *semantic content*?
2.10 What is the essence of model building?
2.11 Explain the *four-universe model* of a computer system!
2.12 What makes a task *simple* or *complex*?
2.13 What do we mean by *emergence*? What are *prior* and *derivative properties*?
2.14 What are the advantages and disadvantages of message-based inter IP-core communication on an MP-SoC (Multiprocessor system on chip)?