GERD DOEBEN-HENISCH

ACTOR ACTOR INTER-ACTION [AAI]

VERSION JUNE 29, 2019 - VERSION 10

UFFMM.ORG

PUBLISHED BY UFFMM.ORG

UFFMM.ORG, ISSN 2567-6458

First printing, May 2019

Contents

List of Figures

Dedicated to those who gave us the prior experience and the inspiring ideas to develop the view offered in this book..

Preface

An AAI Course Program: Within a larger book project about the AAI paradigm represents this text a short, condensed version of the AAI analysis which can be handled within the summer term of a master program. While the larger book project tries to bring together such diverse topics as *Human-Machine Interaction (HMI)*, *Systems Engineering (SE)*, *Artificial Intelligence (AI)*, *Cognitive Science (CogS)* and *Philosophy of Science (PhS)* in one coherent framework called *Actor-Actor Interaction (AAI)*, this shorter text is intended to introduce to a minimal program starting with a problem, analyze the problem in an AAI manner, test the result and stop.

Overview The course follows two main topics: (i) providing the necessary *theory* (ii) to enable a real *analysis process*.

Web Site This small text is located as one sub-topic at the main website `https://www.uffmm.org/`.

Terminology: HMI - AAI - ACI/ ACI In the above mentioned online book the history of the terminology like HCI, HMI, AAI etc. is discussed. In this text – a one-semester course program – the perspective of *Actor-Actor Interaction (AAI)* will be the dominant perspective. But the reader should know that the labeling of *Actor-Cognition Interaction (ACI)* is also valid by pointing to *cognition* as the main factor within the interaction paradigm of actors. One can even go further by emphasizing the dimension of the *distributedness of knowledge* in the different brains of the individual members of a population which can only be *shared and synchronized* by a sufficient *communication*. While the usual communication is the basis for all sharing, new methods of *shared symbolic modeling, interactive simulations* or even *common gaming* can improve this sharing remarkably. These new methods can be understood as an *augmentation* of the classical methods of communication. Thus the acronym *ACI* can have another, more specific meaning.

Part I

Theory

1
Introduction

THE TERM 'ACTOR-ACTOR INTERACTION (AAI)' as used in the title of the book is not yet very common. Better known is the term 'HMI' (Human-Machine Interaction) which again points back to the term 'HCI' (Human-Computer Interaction). Looking to the course of events between 1945 and about 2000 one can observe a steady development of the hardware and the software in many directions.[1]

One can observe an explosion of new applications and usages of computer. This caused a continuous challenge of how human persons can interact with this new technology which has been called in the beginning 'Human Computer Interaction (HCI)'. But with the extension of the applications in nearly all areas of daily life from workplace, factory, to education, health, arts and much more the interaction was no longer restricted to the 'traditional' computer but interaction happened with all kinds of devices which internally or in the background used computer hardware and software. Thus a 'normal' room, a 'normal' street, a 'normal' building, a toy, some furniture, cars, and much more turned into a computerized device with sensors and actuators. At the same time the collaborators of human persons altered to 'intelligent' machines, robots, and smart interfaces. Thus to speak of a 'human user' interacting with a 'technical interface' seems no longer to be appropriate. A more appropriate language game is the new talk of 'interacting actors', which can be sets of different groups of actors interacting in an environment to fulfill a task. Actors are then today biological systems (humans as well as animals) and non-biological systems. Therefor I decided to talk instead of Human-Machine Interaction (HMI) now of 'Actor-Actor Interaction (AAI)'.

[1] For a first introduction see the two human-computer interaction handbooks from 2003 and 2008, and here especially the first chapters dealing explicitly with the history of HCI (cf. Richard W.Pew (2003) , which is citing several papers and books with additional historical investigations (cf. p.2), and Jonathan Grudin (2008) . Another source is the 'HCI Bibliography: Human-Computer Interaction Resources' (see: http://www.hcibib.org/), which has a rich historical section too (see: http://www.hcibib.org/hci-sites/history).

Richard W. Pew. Introduction. Evolution of human-computer interaction: From memex to bluetooth and beyond. In J.A. Jacko and A. Sears, editors, *The Human-Computer Interaction Handbook. Fundamentals, Evolving Technologies, and emerging Applications*. 1 edition, 2003; and Jonathan Grudin. A Moving Target: The Evolution of HCI. In A. Sears and J.A. Jacko, editors, *The Human-Computer Interaction Handbook. Fundamentals, Evolving Technologies, and emerging Applications*. 2 edition, 2008

THE BASIC IDEA OF THE AAI PARADIGM IN THIS BOOK is still centered around a 'concrete interface ($A_{Intf.Real}$)' which allows 'real interaction' with 'real actors (A_{Real})', and these real interfaces have been 'tested' before their usage 'sufficiently well'. Thus the final real interface in real usage has been 'selected' from a finite set of 'real candidates' according to some 'predefined criteria' of 'good usage'.

THE CONTEXT OF AAI is not 'hard-wired' but can be chosen freely. Experience shows us that it is always helpful to fix the conditions under which we want to do our work. What do we presuppose if we start our work? What are our assumptions? What are possible 'frameworks' we are using?

The approach in this book is highly influenced by the paradigm of 'Systems Engineering (SE)' as it is very common in the engineering world. System Engineering can be understood as a bet on the future: given a problem, follow some procedures, and there is some chance, that you will find a solution which can be implemented successfully. The main standards are texts representing the experience of thousands of experts of many thousands of realized projects. What the standards describe is the idealized format of a 'process' with a 'start' and an 'end'. The process is realized by some finite set of 'actors' which coordinate their 'actions' by 'communication', including different kinds of 'artifacts'. We will not speak about systems engineering too much here, but at least let us give a basic idea what it is and how it is related to the main topic 'Actor-Actor Interaction (AAI)' (cf. figure 1.1).[2].

[2] For a first introduction into the idea of *systems engineering (SE)* cf. INCOSE (2015) INCOSE:2015

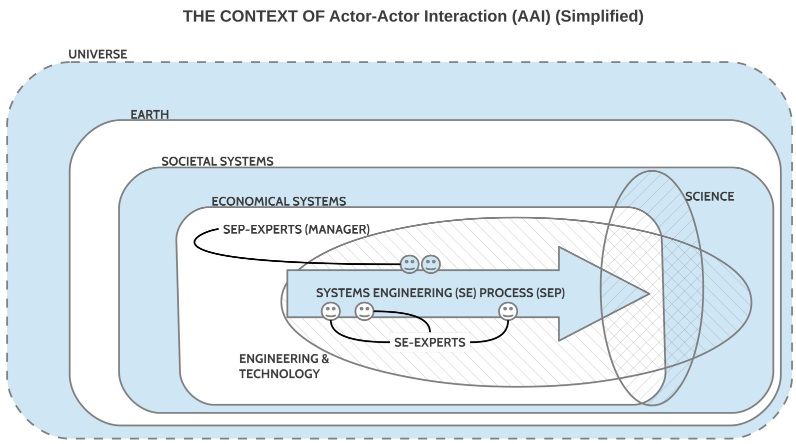

Figure 1.1: A simplified picture of the different contexts for a systems engineering process

'Inside' of a systems engineering process you find different actors called 'experts' which with their experience will drive the process. Outside of the process you have those actors which have to 'manage' the process called 'managers'.

A systems engineering process is always part of some 'economical system' which in turn is part of a 'societal system'. The 'economical system' is the source of many rules for 'how to play the game': available resources, conditions of exchange, gains and losses. Making a systems engineering process an 'economic success' you have to comply with the economic rules. But the economic system is also always interacting with a 'societal system' too: value systems imply preferences and rules to be followed in a variety of different ways, and cultural and human centered patterns will induce additional constraints, which can conflict each other.[3]

[3] Two popular texts which illustrate the interplay of society, technology, and engineering in a broad scope are Eric Schmidt and Jared Cohen (2013) and Yuval Noah Harari (2018) .

Eric Schmidt and Jared Cohen. *The New Digital Age. Reshaping the Future of People, Nations and Business.* John Murray, London (UK), 1 edition, 2013. URL https://www.google.com/search?client=ubuntu&channel=fs&q=eric+schmidt+the+new+digital+age+pdf&ie=utf-8&oe=utf-8; and Yuval Noah Harari. *21 Lessons for the 21st Century*. Spiegel & Grau, Penguin Random House, New York, 2018

Across society and economy we have the realm of 'science' and of 'engineering & technology'. The domain of 'science' manifests themselves as a multitude of distinguished single disciplines whose coherence and unity is only partially in existence. But if you want to know how 'nature' behaves then you have to consult these disciplines. Based on science and as well on collected 'experiences' from many fields and situations we have 'engineering' as a unification of science, craftsmanship, and art, which transforms ideas in working artifacts, which often are machines and whole cities. 'Technology' is one possible outcome of engineering; technology supports the daily life in more and more areas.

Finally, all these mentioned systems are embedded in an overall 'natural system', the earth as part of the universe, inducing many very strong constraints, which to follow is strongly recommended.

To describe this complex matter in detail would burst all boundaries. Therefore we will focus only on that part of the systems engineering process, where AAI comes in and we will thematise the different contexts of a systems engineering process from within the AAI sub-process where it is needed.

THE STRUCTURE OF A SYSTEMS ENGINEERING PROCESS has been described in a formal way by Louwrence Erasmus and Gerd Doeben-Henisch during 2011, when they did some 'conceptual experiments' looking how to formalize a systems engineering process (cf. Erasmus & Doeben-Henisch (2011a/b) [4])

Inspired by modern mathematics (cf. Bourbaki [5]) and the structural approach within philosophy of science (cf. Sneed [6], Balzer et.al. [7]) they pointed out an algebraic structure which can help to describe the elements as well the dynamics of the process . In the following we give a basic description of the main idea restricted to the AAI-analysis phase.

[4] Louwrence Erasmus and Gerd Doeben-Henisch. A theory of the system engineering process. In *9th IEEE AFRICON Conference*. IEEE, 2011a; and Louwrence Erasmus and Gerd Doeben-Henisch. A theory of the system engineering management processes. In *ISEM 2011 International Conference*. ISEM, 2011b. Conference 2011, September 21-23, Stellenbosch, South Africa

[5] N. Bourbaki. *Éléments de Mathématique. Théorie des Ensembles*. Hermann, Paris, 1 edition, 1970

[6] J. D. Sneed. *The Logical Structure of Mathematical Physics*. D.Reidel Publishing Company, Dordrecht - Boston - London, 2 edition, 1979

[7] W. Balzer, C. U. Moulines, and J. D. Sneed. *An Architectonic for Science*. D.Reidel Publishing Company, Dordrecht (NL), 1 edition, 1987

THE AAI-ANALYSIS PART OF A SYSTEMS ENGINEERING PROCESS (SEP) is depicted in the figure 1.2.

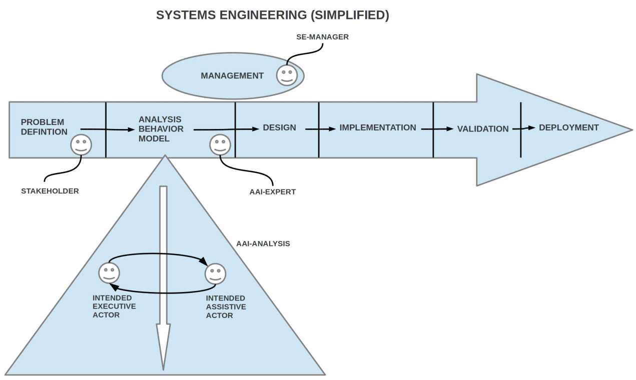

Figure 1.2: Simplified picture of the systems engineering process focussing on the AAI-analysis phase

The AAI-analysis phase is assumed to be 'framed' by a clear beginning and a clear end. The 'beginning' is located in the existence of a 'problem document' D_P, which has been produced by some 'real stakeholder' $A_{SH.Real}$ together with some real AAI-experts $A_{AAI.Real}$; these AAI-experts can be extended by some other real experts $A_{X.Real}$. The problem document D_P describes, what kind of a 'problem' the stakeholder sees and what kind of an 'improvement' he wants. Mostly the 'wishes' of the stakeholder are 'framed' by a set of 'constraints' which have to be matched within the envisaged 'improvements'.[8]

[8] We know that the assumption of a ready made problem document D_P is very strong, because the elaboration of such a document is a real challenge and worth a book on it's own.

THE AAI-ANALYSIS IN A BACKWARD VIEW: Having a 'beginning' of the AAI-analysis and an 'end' one can ask, which steps are necessary to reach the end from the defined beginning? For to do this one can start with the end and asking back: what are the pre-conditions to get the real interface candidates $A_{Intf.Real}$ for the final tests?

Here it is assumed that the 'real interfaces' are 'derived' from symbolically described abstract models of 'assisting actors' ($A.ass$) which are 'used' by some symbolically described abstract 'executing actors' (A_{exec})[9] to fulfill some 'task' (T) within a symbolically describable 'finite sequence of actions' constituting an 'abstract process'; the symbolical description of such an abstract process is called an 'actor story' (AS).

[9] traditionally called 'user'.

Thus, whether the proposed real interfaces are in some sense 'sound' is depending from such an actor story, which describes the intended format of the proposed 'improvements' by taking into account the different constraints mentioned by the stakeholder.

From this follows a very strong assumption implicitly given with this kind of an AAI-approach: the 'problem' (P) described in the problem document D_P can be translated into a sequence of states with at least one start state and at least one goal state, and these states contain intended executive actors A_{exec}, needed assistive actors A_{ass}, a certain 'environment' (ENV) where these processes are assumed to happen, additionally needed 'artifacts' (OBJ), and at least one 'task' (T) which has to be 'fulfilled' by such a process. Possible 'constraints' (C) given as 'non-functional requirements' (NFRs) have to be defined as sets of decidable properties distributed across the different states of the whole process.

That this strong assumption is a 'sound' assumption will be demonstrated in this book. During the course of the arguments you will encounter within the overall AAI-Analysis further special topics like 'Modeling behavior and actors', 'Integrating learning intelligent actors', 'Simulation of actor stories and actor models', 'Automatic verification of non-functional requirements', 'Design of real interfaces', and 'Testing of usability with learning actors and embedded simulations'. Finally you will find several paragraphs pointing to 'philosophical aspects' of this approach which allow a new kind of integration of all these different views.

2
Outline

2.1 Symbolic Space and the Real World

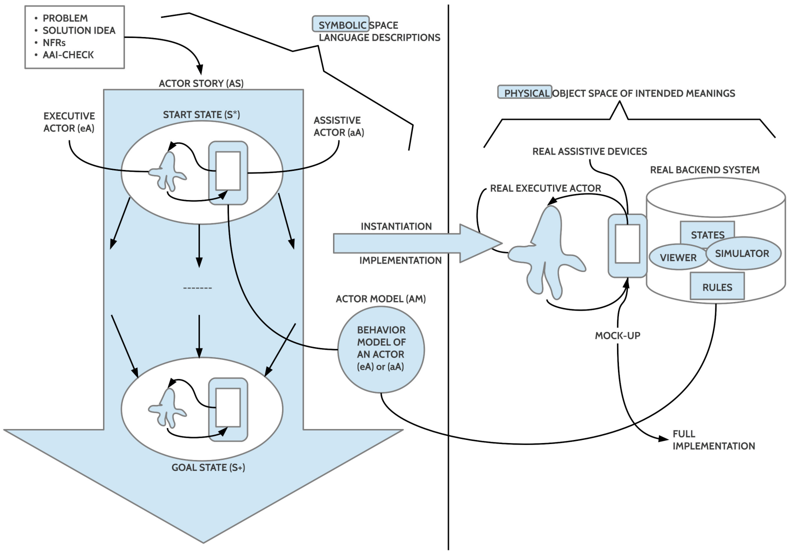

Figure 2.1: The symbolic space of the Actor Story (AS), which then has to become instantiated by real actors with the aid of a real system

Figure 2.1 shows the *symbolic space* of an *actor story (AS)* which has been constructed according to some *stated problem (P)* and an envisioned *solution idea (S+)*. This symbolic space communicates ideas about intended *executing* and *assisting actors (eA, aA)* which are first located in a *start state (S*)* and which can *change* the actual state by doing some actions which cause some change in an actual state generating thereby a *follow-up state (S')*. If one wants to describe the behavior of an actor with more details about the inner structure of an actor then one has to construct additionally an explicit *actor model (AM)* of this actor describing all the known behavior by explaining the internal dynamics.

To check whether these symbolically described possible states of the actor story are working in the *real world (RW)* one has to *instantiate* the intended actors and organize some *test*. This can be done in various *simulations* (including gaming), but the most advanced test will be a *usability test*. In a usability test *real actors* – as close as possible to the finally intended actors – will try to realize the states of an actor story with the aid of a mock-up. A mock-up is a physical device which represents all the important properties of the finally intended assistant actor. The outcome of this usability is either that the symbolic description is fully working in the real world or not. If the test shows deficiencies between the symbolic actor story and the real test then this can reveal some important properties which could be enable a better follow-up test.

2.2 The AS Construction Process

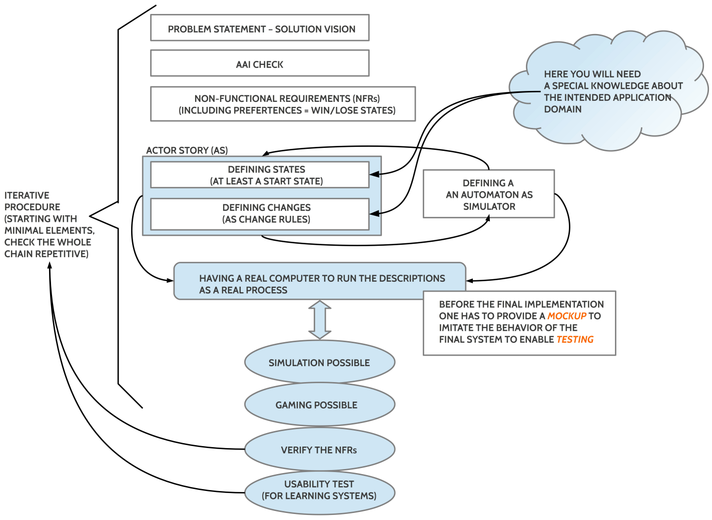

Figure 2.2: Outline of all main elements used in this version of the AAI paradigm

The following text provides an *outline* of all main elements used in an *AAI paradigm*.

All these elements following mainly a sequential procedure. But because this procedure is to a wide extend also an *exploratory* process it is important to *repeat* individual steps or even the whole process if at the end the simulations and/ or tests provide insights in deficiencies. Therefore one has to see this whole sequential process as a *repetitive* process. This recommends to start with as simple as possible assumptions, construct with these assumptions step wise the whole process and get some experience of the effect of all parts working together.

PROBLEM-SOLUTION: Every AAI analysis process presupposes a defined *problem statement* D_p combined with a first idea about a *wanted solution* D_s.

AAI-CHECK: To accept the given problem with the wanted solution one has to check, whether the following *minimal conditions* are fulfilled:

1. The *context (ENV)* of the wanted solution is characterized.

2. There is at least one *task (T)* given which has to be realized within the solution to do the job.

3. There is at least one *executive actor (eA)* which has to fulfill the task as well as at least one *assistive actor (aA)* who shall support the executive actor in doing his job.

NFRs: If the AAI check is positive then one has to give some additional *non-functional requirements (NFRs)* if necessary.

DOMAIN KNOWLEDGE: While the AAI framework as such is a *general framework* intended for all kinds of problems you will need a *special domain knowledge* – often located in so-called *experts* – which allows the inference of the needed facts for the states and the change rules.

ACTOR STORY: To analyze the details of the wanted solution within the intended environment with the implicit tasks and participating actors one has to develop a so-called *actor story (AS)*.

The AS consists of a series of *states (S)* with at least one *start state (S*)* and at least one *goal state (S+)*. A state is a collection of *facts (F)* which can be decided as *true* or not in the assumed environment. Some of the facts describe different *actors (A)* with the executive and the assistive actors as subsets ($eA \cup aA \subseteq A$).

If something is *changing* then a state S.b *before* the change E converts into a *successor* or *follow-up state* S.f. Changes are described by *change rules (X)*. If there exists more than one option to change a state alternatively then the actor story splits up into different lines of state sequences. Possibly these different lines of states can *unify* again at some point later. There can also be a change which effects in some *loop back* if a state has to be repeated again.[1]

[1] A more detailed discussion of 'change' you can find in chapter 5.

CONSTRUCTING SUCCESSOR STATES: In a *first construction phase* the AAI experts have to clarify which are the most important states which have to be assumed to enable an actor story which leads from a start state S* to a goal state S+. And for this they have to identify those *change-rules X* which connect the different identified states. This first construction phase leads to a structure which can *mathematically* be represented as a *graph (G)*. A graph can be turned into an *automaton* which is able to *simulate* this graph G. This gives the foundation for a possible *simulator* σ. And as will be shown later this simulator σ can be built in a general way such that one can implement an appropriate algorithm (software) in a real computer to be able to be used by the AAI experts to simulate any kind of an actor story description.[2]

[2] If a change is triggered by an actor which is *not completely deterministic* than a change can have some variability associated with probabilities which can change. Such a feature turns a complete process into a minimal grade of *uncertainty*. The outcome will not be predictable.

ACTOR AS A LEARNING SYSTEM: In the context of an actor story it is assumed that every actor is principally a *learning system (LS)* with inputs, outputs, internal states as well as a learning function. This leads to the following basic structure of an actor:

$$A(x) \quad iff \quad x = \langle I, O, IS, \phi \rangle \tag{2.1}$$

$$I \quad := \quad Set\ of\ inputs \tag{2.2}$$

$$O \quad := \quad Set\ of\ outputs \tag{2.3}$$

$$IS \quad := \quad Set\ of\ internal\ states \tag{2.4}$$

$$\phi \quad : \quad I \times IS \longmapsto IS \times O \tag{2.5}$$

These assumptions allow for first basic classifications: (i) If the set of internal states (IS) is *empty* or *static*, then the system is principally unable to *learn*. It has to have a completely fixed behavior function which makes the system a *deterministic* system. If there exist internal states *and* these are *changeable*, then the system can principally be a *learning* system, which turns a deterministic behavior function into a *non-deterministic* function.

ACTOR AS ACTOR MODEL: If one wants to describe the details of the learning function of an actor including the details of the main sets {I, O, IS} one has to construct an *actor model (AM)* outside the main actor story. While the actor story is looking to the actors from the *outside* describing how they *behave*, how they *act* in a *situation*[3], an actor model (AM) is looking to an actor from *inside*, from the *internal states* and *processes*[4]

[3] This is called the *3rd person view* by philosophers

[4] This is called *1st person view from philosophers*

INTERFACING AS AND AMs: The *interface* between an *actor story (AS)* and some *actor models (AMs)* is given by the inputs and outputs of an actor. If the actor story describes a certain *action* of an actor, its *output*, then the actor model must *explain* how this output has been generated inside the actor. In the same manner if the actor story describes some *input* to an actor then the actor model must *explain* what happens in the actor on account of such an input. How can an input to an actor influence his output.[5]

[5] For a more detailed discussion see the chapters 4 and 5.

2.3 Testing An Actor Story

If an actor story AS has been constructed one has to check the *cognitive plausibility* of the actor story as well as the *usability* of the intended assistive actors (aAs) by the intended users.

The *cognitive plausibility* is located in the relationship between the *knowledge of the stakeholder* and the possible *experience* when *testing the actor story in a simulation*. If the real experience within a simulation *differs* from the given experience in the brains of the stakeholders than the cognitive plausibility of the actor story is low, eventually too low.

The *usability* of the intended assistive actors (aAs) is located in the relationship between the intended executive actors (eA) and a preliminary mock-up of the intended assistive actors (aA). While the intended executive actor tries to realize a process which is in agreement with the actor story it has to be *empirically measured* (i) to which degree the intended executive actors are *able to realize the actor story* with *this mock-up* and (ii) it should be *subjectively measured* to which degree the intended executive actor is *satisfied* with this process *in an emotional dimension*.

SIMULATION: Having an actor story AS and an *assisting simulator software* σ one can realize a *simulation*, either (i) purely *passive* without interactions or (ii) with *interactions*. In the case of an *interactive simulation* real actors can interact with the simulation and thereby *influence the course of the simulation.* A simulation enables a *shared experience* with a *common understanding* in all participants of the simulation. The *simulation experience* can be compared with the available *real-world experience* of the participants and this allows a special kind of a *cognitive test* revealing those aspects of the simulation which *differ* from the known reality. These *experienced differences* can shed some light on either *deficiencies* of the *simulation* or deficiencies of the *real world situation*.

The introduction of actor models (AMs) simultaneously to an actor story (AS) does not change the concept of a simulation. Actor models occur in the format of a change-rule which in turn is connected to an algorithm which defines its computations.

GAMING: If one extends an interactive simulation with the definition of explicit *win-lose states* then one can turn a simulation into a *game* with real actors which can compete and where some of the participant can become *winners*. Compared to simulations with their somehow *infinite* possibilities identifies a game in advance some *special states of interest* which narrows the scope of the analysis. This helps to focus the test of the process to these special states of interest and enables a much *faster clarification of research questions*. In this sense is *gaming* the *more efficient way of learning* by simulation.

VERIFICATION OF NFRs; ORACLE: If one has defined some NFRs (non-functional requirements) for the actor story then one can after the completion of an actor story including simulation *verify* whether the NFRs are *true* in the actor story with regard to the assumed environment *or not*. A special case of the verification of NFRs is the *oracle* function. Because the verification of NFRs is done in the manner of an *automated prove* with regard to the existence or non-existence of some defined property (associated with a NFR), one can use this mechanism also for to *check* whether a *special state of interest* will *occur or not occur* within a *defined time window* of *all* possible simulations. Such a mechanism can be of great help for the analysis of the possible future of a process, especially without having the need to do *all* the possible (interactive) simulations which is practically impossible on account of the needed time. But because such an oracle-process can only work with the given change-rules *as if these will not change* and without the *non-deterministic behavior of real executive actors* the oracle-results have to be used with caution.

NEED FOR MOCK-UPs: Until that point there exist only *symbolic descriptions* about possible real states. To turn the symbolic descriptions into a *real working system* one has to implement these descriptions into a real system.

But such a full *implementation* is not the job of the AAI analysis. The AAI analysis only examines possible states and possible behavior profiles and checks with the aid of mock-ups whether these ideas will work sufficiently well. *Mock-ups* are physical systems which show all the main physical properties of the intended system without being a full implementation of this system.

USABILITY TESTING: *Usability* reveals something about the way how good the interaction of the intended executive actors with the intended assistive actor works within the whole actor story. Some of the questions which shall be answered by an usability test are: Is it too difficult for the executing actor to *learn* the needed behavior? Does the executing actor need *too much time*? Do continuously occur *too many errors*? To answer these and similar questions one has to prepare a *test scenario* which allows a real executing actor to behave according to the actor story by using the intended assistive actor realized as a *mock-up*. This test has to be managed by a *test coordinator* assisted by some *observing persons* or/ and *recording devices* to produce a *protocol* of the events during the test. The protocols have then to be converted into *test data* which can be used for analytical purposes.

A special point in the AAI usability testing is that within the AAI framework it is generally assumed that the executive actors are by default *learning systems*(which holds for all biological systems). This means that the executive actors eA all have an individual *behavior function* ϕ. This induces within a testing procedure the possible effects that the behavior of a executing actor can change from test to test.[6] To restrict the usability test therefore to only one test run is highly dangerous. It is recommended to *repeat an usability test at least three times.* What number n has to be assumed to be the *optimal* number is still an unanswered question.

[6] Which is indeed the normal case. Therefore you can find in all reports about learning experiments always so-called *learning curves* representing these changes along a time line.

3
Define a Problem

Define a Problem: Because the *space of possible problems and visions* is nearly infinite one has to define as a starting point for a certain process a *problem* together with *a first vision of a 'better state of the affairs'*. This is realized by a description of he problem in a *problem document* D_p as well as in a *vision statement* D_v. Because usually a vision is not without a given context one has to add such an assumed *environment (ENV)* with all the *constraints (C)* which have to be taken into account for the possible solution. Examples of constraints are *non-functional requirements (NFRs)* like 'safety' or 'real time' or 'without barriers' (for handicapped people).

A problem description as well as the first vision of a better state of affairs have to include furthermore at least one *task (TA)* to be fulfilled and some *intended executive actors (eA)* which are *biological* systems; without such biological system there is no need for an AAI analysis.

4
Actor Story

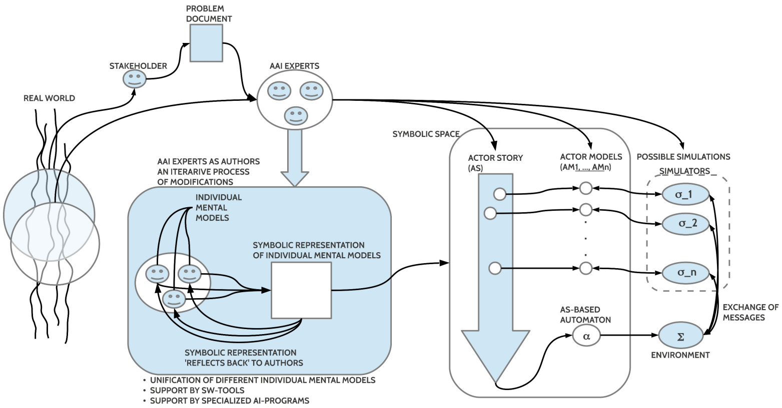

Figure 4.1: Actor stories as well as actor models (see the next chapters) are the product of cognitive processes located inside the participating experts. These AAI experts are interacting with the real world (including the other experts) and construct some mental model of the world and the problem to be solved. This model is converted into a symbolic representation understood as actor story (AS) and possibly actor models (AMs).

OBJECTIVE FOR AN ACTOR STORY: As outlined in the figure 4.1 the general process of generating an actor story (AS) – and in later chapters also actor models (AMs) – is rather complex with a strong *cognitive component* which is located in the inner cognitive processes of the participating AAI experts. The existence of a problem document D_p and a first vision document D_v (cf. chapter 3) offers for the generation of an actor story primary points of reference.

Within an *actor actor interaction (AAI) analysis* it will be analyzed how the vision can be realized. It is assumed in this text that the realization of the vision assumes a situation in which at least one *executive actor (eA)* and an *assisting Actor (aA)* supporting the executive actor has to do some job which requires to fulfill some task. This implies that there is more than one state which has to be passed to reach the fulfillment in some *goal state (S+)*. Therefore a minimal *process* has to happen beginning with a *start state (S0)* and ending up in some *goal state (S+)*. Besides the actors there can be other possibly other *objects*, all associated with *properties* and eventually embedded in *relations*.

The internal *mental models* have to be represented externally by *symbolic expressions* of some language L. Those symbolic expressions representing objects with properties or objects within a relation are called *statements* describing *facts* which can be decided as either (i) *corresponding* to some real fact, (ii) *not corresponding* to real facts or (iii) actual *undefined* because there is no real situation available for comparisons. One can also use *grades of similarity* which allow *fuzzy* relationships between mental facts and real facts.

The case of *correspondence* is often described as 'the statement is *true*' and the case of *non-correspondence* is described as 'the statement is *false*'.

For convenience in this text a state S is called a *set of facts (F)* although the state S is only a set of *expressions* called *statements* and these statements are within the cognition of an expert mapped into facts which can be distinguished as *mental facts (F.m)* from mental representations of *empirical facts (F.e)*. Thus a *correspondence* is here assumed to be a cognitive relation between two different kinds of mental representations: those (F.e), which are caused by the perception (and cognitive processing) of assumed empirical facts, and those (F.m) which are produced by cognitive processes only. These mental facts *can* correspond to some empirical facts (F.e), but whether this is the case is completely arbitrary.

Every *situation/ state* is here assumed to be *static*. In that moment where a *change* occurs it is assumed that at least one fact (f) of the set of facts (F) has been changed, either by *deletion (-F)* or by *creation (+F)*. Thus one can speak of the *effect (E.f)* of the change between the state before (S.b) and the state following (S.f) as the *unification* of the deleted and created facts: $E.f = -F \cup +F$. The following state S.f is then the outcome of the operation $S.f = S.b - (-F) + (+F)$.

KINDS OF SYMBOLIC EXPRESSIONS: The process of translating *cognitive* representations into *symbolic* expressions is open for a great variety of expressions. In this text there are two favored kinds of symbolic representation: (i) using *everyday language* L_0 (in this text English), and (ii) *mathematical language* L_m. These two basic kinds can be extended on demand by (iii) a *pictorial language* $L_{pict.0}$ mimicking the everyday perspective like a comic-strip or (iv) by a *pictorial language* $L_{pict.m}$ which visualizes the structure of states (Doeben-Henisch & Wagner (2007)[1]).

It has to be kept in mind that the symbolic expressions as such are *meaningless*! They receive their possible *meaning* within those cognitive processes which are *mapping* different kinds of *mental structures (T.m)* into a set of *symbolic expressions (L)*. If we call this mapping the *meaning function* μ of the language L, written as $\mu : T.m \longleftrightarrow L$, then $\mu(L)$ produces the *meaning* of the expressions of the language L and $\mu(T.m)$ produces the language expressions which are used to *encode* the meaning by these expressions.

SEMIOTIC ACTORS: This language game of expressions as elements of a language and meaning associated with these expressions encoded in the internal states of an actor leads to a minimal theory of language usage, which traditionally is handled within *semiotics* (For a good overview of the whole field of semiotics see Nöth (1990, 2000) [2]). I have transformed some of the classical approaches – especially that from Charles Morris (1938, 1971)[3] – into a concept of the semiotic actor extended by a framework for cognitive processes enabling mental structures (cf. Doeben-Henisch (1998,

[1] G. Doeben-Henisch and M. Wagner. Validation within safety critical systems engineering from a computational semiotics point of view. *Proceedings of the IEEE Africon2007 Conference*, pages Pages: 1 – 7, 2007. DOI: 10.1109/AFRICON.2007.4401588

[2] Winfried Nöth. *Handbook of Semiotics*. Indiana University Press, Bolomington - Indianapolis, 1 edition, 1990. Enlarged and completely rewritten edition of the 'Handbuch der Semiotik' (1985); and Winfried Nöth. *Handbuch der Semiotik*. J.B.Metzler, Stuttgart - Weimar, 2nd edition, 2000. Completely rewritten 2nd edition of the 'Handbook of Semiotics' (1990)

[3] Charles W. Morris. Foundations of the Theory of Signs. volume 1 of *International encyclopedia of unified science*, pages 1–59. The University of Chicago Press., Chicago (Illinois), 1938. ISBN 0226575772; and Charles Morris. *General Theory of Signs*. Mouton, Paris, 1971

2007)[4]).

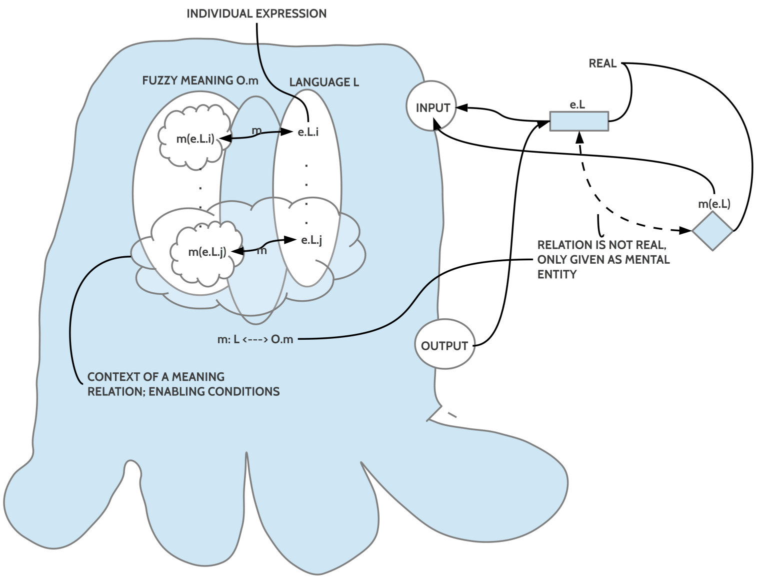

Figure 4.2: Minimal structure of a semiotic actor

[4] Gerd Doeben-Henisch. Semiotic Machines. In *Signs and Space - Raum und Zeichen. An International Conference on the Semiotics of Space and Culture*, pages 313–327, Tübingen (DE), 1998. Gunter Narr Verlag; and Gerd Doeben-Henisch. Reconstructing human intelligence within computational sciences: An introductory essay. In A. Loula, R. Gudwin, and J. Queiroz, editors, *Artificial Cognition Systems*, pages 106–139, Hershey - London - Melbourne - Singapore, 2007. Idea Group Publishing

Figure 4.2 provides a basic outline of this actor based *cognitive semiotic framework*. The main idea is given in the assumption that the *inner states* of a semiotic actor contain different kinds of *mental* structures which are processed by the *brain*. And to distinguish those brain processes which are related to mental facts from all the others (no sharp boundaries!) this subset of brain processes is called *cognitive space*. The main components of this cognitive space are processes related to *input (perception)*, to *output*, to representatives of *symbolic expressions* as well as representatives of *meaning correlates*, as well as the overall *meaning function* mapping symbolic expressions into meaning correlates and vice versa. The meaning correlates are assumed to be *fuzzy* and *dynamic* structures with many inter-dependencies. Furthermore it is assumed that meaning functions are always embedded in some *context* which implicitly defines different kinds of *conditions* which have to be taken into account before a meaning function will *match* a *real* situation *external* to the inner states. Thus if there occurs an external (real) symbolic expression e.L of some known language L then a semiotic actor which has learned the language L has some probability that his internal states *automatically* (unconscious) *activate* some meaning functions which in turn activate possible mental meaning correlates and 'in the light' of these activated meaning correlates the semiotic actor perhaps can *identify* some real matter – including its context – which matches with the activated mental structure sufficiently well. In that case the semiotic actor *interprets* the matched meaning correlate as the *intended meaning* of the symbolic expressions. This means that for the semiotic actor there 'exists' a meaning function 'in his head' but not in the external reality. Those semiotic actors which do not know the language L will not be able to 'see' this meaning function. This everyday fact reveals the eminent constructive part of available knowledge to see *different realities* whereby there is *only one* real world.

MATCHING REALITY: Every usage of a language whose expressions are assumed to correlate somehow with a known meaning which to some degree is also related to the external, empirical world has to deal with the *matching* of *internal, mental structures* – the assumed encoded meaning correlates – and some parts of the real world. Otherwise symbolic language would not be useful for *communication* and communication mediated *cooperation*. This empirical aspect of everyday language usage will in this text be assumed to be a basic feature of the usage of some minimal formal language L_m.

We assume a minimal formal language $L_{m.0}$ with the following elements:

$$\begin{aligned}
L_{m.0}(L) &\quad iff \quad L = \langle E_{Obj}, E_{Prop}, E_{Rel}, F, T\rangle \qquad (4.1)\\
E_{Obj} &\quad := \quad \text{Object names}\\
E_{Prop} &\quad := \quad \text{Property names}\\
E_{Rel} &\quad := \quad \text{Relation names}\\
F &\quad := \quad \text{Fact statements}\\
E_{Prop} \times E_{Obj} &\quad \subseteq \quad F\\
E_{Rel} \times (E_{Obj})^n &\quad \subseteq \quad F\\
T &\quad := \quad \text{Text}\\
2^F &\quad \subseteq \quad T
\end{aligned}$$

A simple example: We assume as a simple text T1 the following set of expressions: {HOUSE(H), DOOR(D), PERSON(P), OPEN(D) PART-OF(D,H)}. An assumed translation into everyday language could go as follows: There is an object with name 'H' which has the property to be a house; another object with name 'D' which has simultaneously the property to be a door and the property to be open; another object with name 'P' and the property to be a person, and finally it is stated that the two objects with the names D and H are embedded in a relation 'PART-OF', i.e. the object with name D is assumed to be part of the object with the name H.

To apply a minimal formal language L.m.0 to reality we need a *non empty population of semiotic actors A.sem* which have learned to use the minimal formal language $L_{m.0}$ with an *internal encoding for meaning correlates* connected to the expressions.[5] Furthermore it has to be assumed that semiotic actors are always part of some *real world situation*.

If a semiotic actor A would utter (or write or ...) the text T1 which shall describe a real situation S1 then we can distinguish the following matching cases:

1. There is according to the *learned meaning function* μ of the language L.m.0 a *match* between text T1 and the intended real world situation S1 which is given as a *perception*, i.e. there is a house with an open door and a person within this perception. In this case one would call such a statement with regard to the mental relation between *known = mental* representation and *perceived = real* representation a *true* statement.

2. Parts of the *real = perceived* situation S1 *do not match*, e.g. the perceived door is closed although the text states that the door is open. If one

[5] This encoding for meaning correlates is here called the *meaning function* $\mu_{L.m.0}$. The *used* language L as well as the presupposed *meaning function* μ can be used to define a *language community* $COM_{L,\mu}$.

assumes that the property of being *open* is the logical negation of being *closed* then one would say that this statement is *false*.

3. The actual real situation where the semiotic actor is located has no real counterpart to the encoded meaning of the text T1. In this case the semiotic actor cannot decide whether the text T1 is true or false; in that case the text is *undefined*.

While these three cases are really base cases there exist in everyday communication many variants of these cases. Two main versions are mentioned here:

1. Very often one does *not* use such *clear cut cases* as mentioned above but one works with some *grades of similarity*. Thus the one object with the name 'P' is associated with the property 'PERSON' but this association is only stated with a similarity of say 75%. There are some aspects in the appearance of this object which do not fit completely with the associated property.
2. If an object is associated with a property or a relation which can *change* then such a change is often not precisely predictable but is associated with some *probability* π. Therefore if the object with name 'D' is at a certain point of time t associated with the property 'OPEN' then there exists some probability π that this property can be substituted by the property being 'CLOSED'. This probability can further be associated with certain *conditions* which themselves can be connected to properties.

Having these aspects of how a text can match the reality (true, false, undefined, grades of similarity, probability of change), then one can imagine that these different aspects can be mixed up in many ways. A probability of change for the property of OPEN to become CLOSED can also be intermixed with some percentages of similarity in the sense that being CLOSED or being OPEN can occur in a 'graded way: not completely closed, not complete open, etc.

GENERATING FOLLOW UP STATES: If one understands how a semiotic actor A can construct a text T by using a language L with an associated meaning function μ then one can understand how a semiotic actor can define a *possible situation* by writing such a text T. Every member of the same *language community L* can take this text T and can check whether this matches some real situation and how. Instead of describing only given or past situations a semiotic actor can also use such a text T to describe a *possible future situation* and it is the task of experts to *find possible realizable changes* which can be applied on the actual situation S* in a way that it generates one – or more – new situations $S+_i$ in the future to become *real*.

This induces implicitly the dimension of *time*: to speak of *now* and *after* or *follow up* or *past* presupposes that every participating expert actor can distinguish in his cognitive space such *order relations* interpretable as representations of some real occurrences. While the *reality* is always a *NOW* for every kind of an actor it has been shown to be constantly changing.

This constant changes can only be *detected as changes* while the cognitive space of an actor can *store* somehow a 'now', can *remember* somehow stored items, can *compare* – mostly automatic (unconscious) – stored items with actual (now) items and can draw some *conclusions* from these comparisons, e.g. establish an *ordering relation* as 'BEFORE(A,B)' or 'AFTER(B,A)' etc. The old habit of using *nature-based cyclic changes* like day-night cycles has been enriched in modern times by *mechanical clocks* which generate with appropriate precision *time events* – often called 'ticks' – which are mapped into *number signs* which then can be used to label these ticks like {1,2,3, ...} or {07:27:33} or {June 28, 2019} etc.[6]

[6] For a basic introduction into the phenomenon of time see Whitrow (1980) and Audoin & Guinot (2001) .

G.J. Whitrode. *The Natural Philosophy of Time*. Clarendon Press, Oxford, 2nd edition, 1980. The 1st edition of the book appeared 1961; and Claude Audoib and Bernard Guinot. *The Measurement of Time. Time, Frequency and the Atomic Clock*. Cambridge University Press, Cambridge - New York - Melbourne, 1st edition, 2000. The original French edition of the book appeared 1998

CHANGE: Now, if there is a dimension of *time* available with the timely ordering for a state S being *before, after* or *equal* to another state S', one can define a general concept of *change* based on the available fact-statements F of a given state S: a change X from given state S to some follow-up state S' is defined by a set of fact-statements named -F which has been *deleted* going from S to S' or/and a set of fact-statements named +F which has been *added* in S' compared to S. If one calls the set {-F, +F} the *effect set X.e* then a change contains at least a *non-empty effect set X.e*. If T is the text describing the state S then one can construct the new text T' describing the follow-up state S' with the operation: $T' = T - (-F) + (+F)$.

Usually a change X happens not 'from nothing' but is associated with a *condition (X.c)* which has to be given that a change takes effect. The condition X.c is nothing else as some subset of the given state S written as $X_C \subseteq T$. Another aspect is usually that an effect X.e will *not necessarily* follow the fulfillment of a condition X.c but restricted to some *probability* $\pi \in \Pi$. Thus we have the configuration, that the effect X.e of a change X will take effect with some probability π_i if a certain condition X.c will be fulfilled in the actual state S.

GENERATING AN ACTOR STORY (AS): Putting all these elements together it is only a small step to a complete *actor story (AS)*. From a formal point of view one can use the mathematical concept of a *graph* Γ which will be expanded by some additional properties.

An ordinary mathematical graph is defined as follows:

$$\begin{aligned} \Gamma(g) \quad & iff \quad g = \langle V, E \rangle \\ V \quad & := \quad vertices \\ E \quad & := \quad edges \\ E \quad & \subseteq \quad V \times V \end{aligned} \tag{4.2}$$

Related to the case of an actor story (AS) the *vertices* have to take the role of the situations (or states), and the *edges* represent the transitions from one state S to the follow-up state S'. To include this additional information one has to install first a mapping from vertices V into the set of fact-statements F, written as:

$$\lambda \quad : \quad V \longrightarrow 2^F \tag{4.3}$$

Thus, having some vertex $v \in V$ the expression $\lambda(v) = T$ and T is a subset of 2^F.

To include the change-descriptions X associated with probabilities Π into the transitions one can proceed either by changing the definition of an edge or one can analogously to the facts establish a new mapping from edges into change descriptions:

$$\epsilon \quad : \quad E \longrightarrow X_c \times \Pi \times X_e \tag{4.4}$$

Thus having an edge $e = (v, v') \in E$ with the change-description $\langle X_c \times \Pi \times X_e \rangle$ then the text $T = \lambda(v)$ has to have the change-condition X_c as a subset to activate the possible change X_e with probability Π. The follow-up text $T' = \lambda(v')$ can then be computed by the operation $T' = T \cup X_e$ with $X_e = \{-F, +F\}$.

Putting all pieces together we get the extended definition of an *actor-story graph* Γ_{AS} as follows:

$$\begin{aligned}
\Gamma_{AS}(g) \quad & iff \quad g = \langle V, E, F, X, X_c, X_e, -F, +F, \Pi, \lambda, \epsilon \rangle \\
V \quad & := \quad vertices \\
E \quad & := \quad edges \\
E \quad & \subseteq \quad V \times V \\
F \quad & := \quad Fact - statements \\
\{-F, +F\} \quad & \subseteq \quad F \\
X \quad & = \quad (X_c, X_e) \\
X_c \quad & := \quad Condition\ part \\
X_e \quad & := \quad Effect\ part \\
X_e \quad & = \quad \{-F, +F\} \\
\lambda \quad & : \quad V \longrightarrow 2^F \\
\epsilon \quad & : \quad E \longrightarrow X_c \times \Pi \times X_e
\end{aligned} \tag{4.5}$$

Thus an actor story (AS) is basically a graph whose vertices are associated with fact-statements constituting a text which represents some potential state/ situation, and edges which are labeled with change-descriptions $\langle X_c, \pi, X_e \rangle \in X$ which determine under which condition X_c with which probability π a certain effect X_e will happen. Because *time* is another important aspect of change one can include explicitly some information of the duration until an effect can begin and will end.

ACTOR STORY PROCESSING: The definitions of an actor story so far describe only the descriptions of states or intended changes, but they do not explicitly define an operator which takes texts and changes as input and generates the follow-up text as output. In this text this *processing knowledge* has been described rather informal. To make it explicit one has to define an *automaton* α to do this job. This will be done later under the heading of *simulation*.

UNIFYING STATES: If one defines *different* actor stories with different sets of states and edges[7] then the question can arise how one can *synchronize* these different subsystems. There are some cases to distinguish:

[7] Think about a city with different subsystems for demography, budget, water supply etc.

1. If there are n-many different states to unify then one declares a new *super-state* where all the other states are *sub-states*.
2. If there are no relations between the sub-states then nothing else will happen. Every sub-state will be processed with its own change-rules as before.
3. If there shall exist a new relation R between two before different states, then there must in every participating state of the relation a variable be created which will be part of the relation. Change rules can then become influential to another state if the new relation R makes an influence explicit. **Example**: If one actor story AS1 deals with the population dynamics of a city with a population POP, the birth-rate BR and the death-rate DR, and another actor story AS2 deals with the water-supply for this city with the actual water reservoir WR, the possible input to this water reservoir from some spring SPR, and the water consumption of the city WCN. Unifying both systems would require to relate the population POP with the water consumption WCN by some new mapping like wcnpop(POP)=WCN. For to extend the two old actor stories AS1 and AS2 to a new unified story AS12 = AS1 $\cup$ AS2 one has then only to add some new change-rule wcnpop() to the unified list of change rules X12 = X1 $\cup$ X2 $\cup$ {wcnpop()}.

From this follows that the unification of before separated actor stories AS1 and AS2 requires in the worst case the introduction of new change-rules associating two before unconnected variables with a new function. This induces a *cognitive enrichment* of both actor stories in the unified version.

5 Actor Model Embedding

ACTOR MODEL EMBEDDING: In the preceding chapter 4 about the basic elements of an actor story (AS) one can talk about states as sets of facts and some of these facts can be understood as facts describing an object which has some kinds of perceptions as well as actions, but this 3rd-person view of an actor story does not allow for descriptions about the *inner states (IS)* of such an object which could *explain* the observable behavior. To talk about such inner states one has to define a separate *actor model (AM)*. To make such potential actor models *work* in interaction with an actor story one has to define this interaction in a precise way. The primary interface for such an *actor story* - *actor model interaction* are the *changes* which turn a *before-state S.b* into *follow-up state S.f*. Such a *change* is defined by the change of at least one fact f which either will be *deleted* from S.b to S.f or will be *created* from S.b to S.f.

5.1 Rewriting the Change as an Actor

LOCATE AN ACTOR WITHIN A CHANGE: To understand the interaction of actors in connection with an actor story one must understand the structure of an observable change between two states.

CHANGE AS OBSERVABLE EFFECT: As described in the chapter 4 about the actor story (AS) an actor story can be understood as a directed graph whose nodes are states as sets of facts and the connecting directed edges are possible changes. A *change (X)* is described by the differences in the sets of facts between the *before-state (S.b)* and the *follow-up state (S.f)*. With regard to change the following cases are possible: (i) a fact F from the before-state S.b will *disappear* in the follow-up state S.f represented as *-F* or (ii) a fact F in the follow-up state *is new* compared to the before-state represented as *+F*. This set of *deleted facts -F* together with the *newly created facts +F* is called the *effect of the change (N.e)* with N.e= {-F, +F}. As soon as one can identify some effect N.e one can look for a possible *source (N.s)* in the realm of the before-state. A source N.s is a subset of the facts of S.b which can be identified as preceding the observable effect. Having a possible source N.s then one can observe with some probability p.i that the observable fact will occur within a certain time-frame Δ defined by two time points (t,t') (cf. figure 5.1).

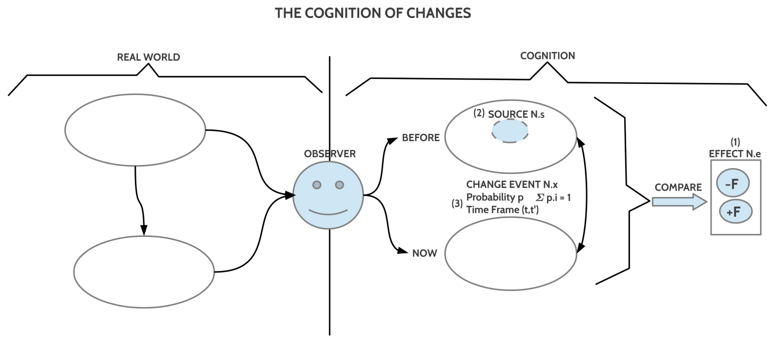

Figure 5.1: The cognition of observable changes

THE COGNITION OF CHANGES: If one tries to analyze the *language game* associated with observable changes then one encounters some difficulties. While an object *o* usually has some *permanence* one can talk about such an object *o* by pointing to this object and using names N.o. If a change happens, the *previous state S.b* which did change will *disappear* in its original format and will be *replaced* by a *follow-up state S.f* revealing something *new*. In everyday language we have no problem to talk about changes with appropriate names N.x similar to talking about objects, but looking closer one can detect a real difference: although we assume that the new follow-up state S.f can be recognized as new because he is *different* to the preceding state S.b, in the real world this difference is not present. If one assumes that every observer has *inner states* which enable some *memory* with the additional capability to *remember* stored items and being able to *compare* remembered items with new, present items, then one can explain that we can talk about changes and being able to name the differences *based on these cognitive representations and operations*. This is depicted in the right half of the figure 5.1. The figure shows a simple model of a minimal cognitive structure with the elements (i) *storing* perceptions as re-callable items; (ii) being able to *compare* stored items with regard to possible differences; (iii) *identifying possible sources* for such effects together with probabilities as well as probable time frames for the occurrence of the effect.

If one makes the assumption that the identified probability p is 'part of a probability space' with $sum(p_i) = 1$ then one has to assume that a source N.s can be associated with different kinds of effects $\{N.e_1, ..., N.x_k\}$ which are exclusive. This means that an actor story with an identified source N.x having different probabilities p_i can have different follow-up states $\{S.f_1, ..., S.f_k\}$; thus the path containing the state S with the source N.s will be splitted up into different continuations.

The process of the identification of a possible source N.s for an observed effect N.x hides another cognitive property, that of *pre-knowledge*. To state that some facts are indeed an *effect N.x* and not only some kind of a *difference* presupposes that one is able to *embed* observed differences in some *cognitive (=abstract) relation* which relates the observed differences to some source N.x. Such a relation is a cognitive fact which belongs to what usually is called *knowledge*, which is assumed to be located in the inner states of an observer. Without such a knowledge there wouldn't exist relations and without relations there wouldn't it be possible to identify a source for some differences turning the differences into a possible effect. Thus the *detection* of something as being a possible *source* for an observed effect is completely depending from a presupposed knowledge. Science has many examples for detections of differences as effects of some presupposed sources.[1]

[1] One of many examples in science: Only in 1964 it happened that two American radio astronomers detected signals in their data (= the differences, which can become effects) which they discussed with their colleagues and then came to the conclusion (= because of presupposed knowledge about possible relations) to *interpret* the signals as the *cosmic microwave background (CMB, CMBR)* (= the differences became effects within a relation) , which could be a remnant from an early stage of the universe (= the possible source of the effect), also known as relic radiation. This interpretation presupposed as knowledge a complex physical theory about the development of the universe (cf. PenziasWilson:1965).

MULTIPLE SOURCES: If there exists only one source N.s then possible *different* effects N.x are exclusive: only one of the possible effects can happen at the same time. Nevertheless the whole actor story AS has to be splitted up from that state onward which shows these alternatives. But there can be more than one source in one state $\{N.s_1, ..., N.s_k\}$ and every source $N.s_i$ can have its own special effects $N.e_i$. While for each single source $N.s_i$ there can only one of many effects happen at the same time, each source $N.s_i$ can generate one effect $N.e_i$, and *these different effects are simultaneous*! And because a real situation does not split up into alternatives *all* these effects have to be *unified* in *one follow-up state* $S.f$.

This induces the question how one can unify more than one effect in one follow-up state S.f?

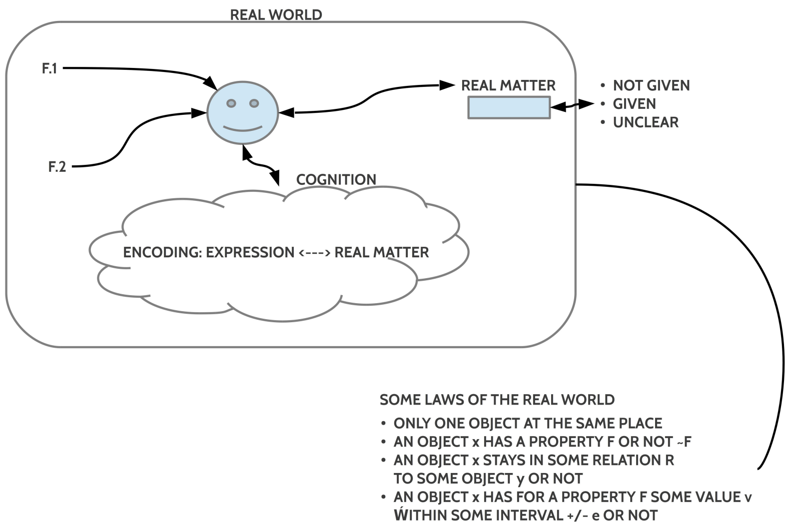

Figure 5.2: Condition for unifying different effects in one state

One possible direction for an answer is the *meaning dimension* of the used expressions. Every source N.s is a set of *expressions* where each expression is classified as a *fact*. In the standard situation the AAI expert has in his cognitive space an *encoding schema* translating a fact-expression F into some *intended matter* which realizes the possible meaning of the expression. But there is a difference: the intended meaning in the *cognitive space M.cog(F)* and the meaning in the *real world M.real(F)*. There are three basic cases: (i) M.real(F) matches the intended meaning M.cog(F) (then the expression is called *true*, indicated often as '1'); (ii) M.real(F) contradicts the intended meaning M.cog(F) directly (then the expression is called *false*, indicated often as '0'); (iii) The relation between the intended meaning M.cog(F) to the real world is unclear, because there is no matter, which seems to correspond to M.real(F). Then the status of the expressions F in the real world is *undefined*.[2]

Within these basic cases of being *true, false or undefined* there are some more detailed cases possible. In the real world one has identified some basic *laws*, which define some *constraints* for the matching of expressions to matter. Here some basic cases:

1. No two different objects can occupy the same space at the same time.
2. An object x can not have at the same time property F as well as $\neg F$.
3. An object x can not stay at the same time in a relation R to some other object y and not $\neg R$.
4. An object x can not have for a property F within a defined interval +/- ϵ the value v and not.

From this follows that the facts which are part of an effect N.e of some source N.s have to be clarified with regard to these *truth conditions* of the presupposed real world. If one assumes that between to effects $N.e_i$ and $N.e_j$ is always a minimal *time delay* such that the one effect is *earlier* then the other effect then the realization of the earlier effect *comes first*. Nevertheless it has to be defined what happens if one effect *touches* another effect *some time later*. In many real world situations the hitting of one object by another is not only possible but often intended (some kinds of sports, accidents, battle situations in war, etc.).

VIRTUAL WORLDS: In the preceding section only the case of a real world and of the cognitive space of an expert living in the real word is assumed. In science, education, engineering, and different kinds of training *parts of the real world* are *substituted* by a *model world* which mimics important aspects of the real world or plays with a *fantasy world* to explore new dimensions. In these cases the *meaning of the real world* M_{real} has to be substituted by a constructed *artificial meaning relation* $M_{virtual}$.

SOURCE AS AN ACTOR: An important case of a special format of a source is a source structured as an *input-output system* representing an *actor (A)*. An actor includes a *behavior function* ϕ which defines the *output (O)* as response to the *input (I)* with some sensitivity for the *internal states (IS)*, which can change, written as $\phi : I \times IS \longmapsto IS \times O$

Depending from these basic assumptions about an actor one has to require that a *source N.s* which shall be recognized as an actor must consist of a subset of the facts $F_{S.b}$ of the state-before S.b, which can be divided into three further subsets: (i) there is a subset representing an *input-output system* assumed to be the *actor* as an object; (ii) there is another subset of facts which are presupposed as an *input* to the actor; (iii) there is a further subset of facts related to those facts, which can become changed by the actor as the actors *output*. The actors output then will be assumed *to trigger the observable effect*.

As it is known from the real world with the biological systems the same part of the environment – a set of facts as possible input – can be perceived in a different way by different kinds of biological actors. The same holds for

[2] In modern logic such a 3-valued truth system has been extended to many-valued cases or so-called *fuzzy* systems. But this does not change the basic structure. It gives only more flexibility in practical applications.

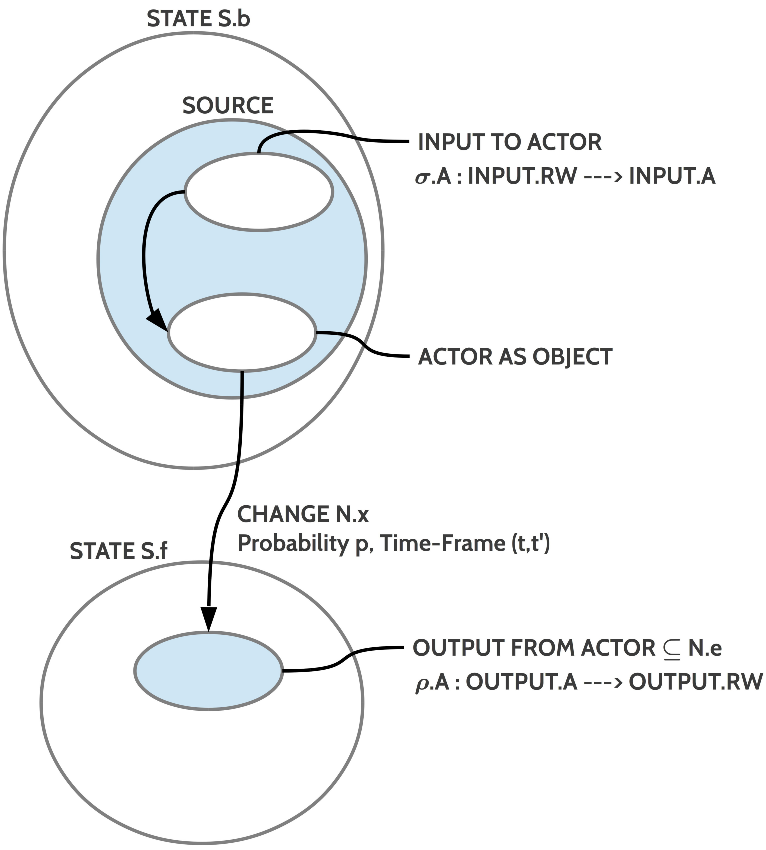

Figure 5.3: Source is an actor with input and output

different kinds of robots. This repeats for the output of a system which can have different effects for the environment.

To take this individual conditions into account it will here be assumed that for every actor there exists an *individual input sensor function* σ mapping the subset of facts representing the possible input I_{RW} into the perceived input I_A of the perceiving actor, written as $\sigma_A : I_{RW} \longmapsto I_A$. Similarly there exists a *typical output function* ρ which translates the individual output of an actor O_A into a corresponding set of *output facts* O_{RW}, written as $\rho_{RW} : O_A \longmapsto O_{RW}$. Examples from the real world are the way how the same movement of a body will cause completely different effects depending whether the movement has been done in the inter-planetary space, in the air on the surface of the earth or under water.

MULTIPLE ACTORS: As in the case of a change with a general source it is possible that there occurs more than one actor in a state. Similarly to the source case do different actors as part of a source create different possibilities of effects which have to be handled *simultaneously*. And as in the general case of a source the different effects of the different actors have to become *unified* in one follow-up state S.f.

5.2 How to Apply Changes?

After the introduction of the general concept of a source as part of a change and then of the special case of a source which has the format of an actor let us have a look how to apply these concepts.

1. To generate a *follow-up state S.f* for a *before.state S.b* we have to identify at least a *source N.s* and a possible *effect N.e* with *probability p* and an associated *time-frame (t,t')*. For this we could write as a *change-rule N.x*: $\langle S.b, S.f, N.s, p, (t, t'), N.e \rangle$
2. If there are multiple effects $\{N.e_1, ..., N.e_k\}$ associated with *one* source N.s (obeying the constraint $sum(p_i) = 1$)) then the actor story has to be splitted after the before-state S.b. Thus one has to write down multiple change-rules $\{N.x_1, ..., N.x_m\}$.
3. If there are multiple sources $\{N.s_1, ..., N.s_k\}$ in *one* before-state S.b then it can happen that each source triggers an effect $\{N.e_1, ..., N.e_k\}$ and then all these effects have to be unified in the one follow-up state S.f. In this case the change rule N.x constitutes a whole set of rules like $\{\langle S.b, S.f, N.s_1, p, (t, t'), N.e\rangle, ..., \langle S.b, S.f, N.s_k, p, (t, t'), N.e\rangle\}$ with different kinds of probabilities, time-frames, and effects for each source $N.s_i$. A shortened version would read as $\langle S.b, S.f, \langle N.s_1, p, (t, t'), N.e\rangle, ..., \langle N.s_k, p, (t, t'), N.e\rangle\rangle$
4. Using the special case of a source as an *actor* then one has to specify additional subsets in the following way: besides the before-state S.b and the follow-up state S.f, the source N.s has to be divided into an input-set $I_{S.b}$ with the translator function $\sigma_A(I_{S.b}) = I_A$, the actor-object A, a probability p with a time-frame, a translator function $\rho_{S.b}$ of the environment to translate the actor output O into the effect N.e, which can be written: $\langle S.b, S.f, \langle \sigma_A(I_{S.b}) = I_A, A\rangle, p, (t, t'), \rho_{S.b}(O) = N.e\rangle$.

5. If an actor has more than one possible output it has to be handled as the case of a source with multiple different effects, i.e. the one before-state S.b has several follow-up states each with another change-rule N.x.
6. Finally, if there exists more than one actor then there exists only one follow-up state S.f – as in the case of multiple sources – but there is a whole set of change rules whose effects have to be unified.

5.3 Actor as a Learning System

ACTOR AS A LEARNING SYSTEM: In the context of an actor story it is assumed that every actor is principally a *learning system (LS)* with inputs, outputs, internal states as well as a learning function. This induces that an actor can be represented as a change-rule whose actions can cause a state-change depending from the input of the actor in an actual state.

ACTOR AS ACTOR MODEL: If one wants to describe the details of the learning function ϕ of an actor including the details of the main sets {I, O, IS} one has to construct an *actor model (AM)* outside the main actor story. While the actor story is looking to the actors from the *outside* describing how they *behave*, how they *act* in a *situation*[3], an actor model (AM) is looking to an actor from *inside*, from the *internal states* and *processes*[4]

[3] This is called the *3rd person view* by philosophers

[4] This is called *1st person view from philosophers*

6

Dynamic AS and AMs Interactions

DYNAMIC ACTORS: It has been already stated that in the context of an actor story it is assumed that every actor is principally a *learning system (LS)* with inputs, outputs, internal states as well as a learning function. If one indeed has really learning systems, then this has far-reaching consequences for the course of an actor story.

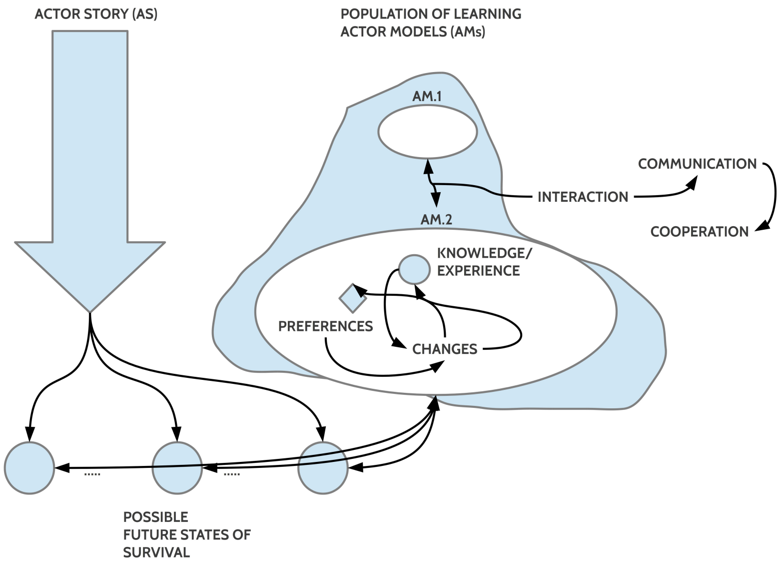

Figure 6.1: Outline of a dynamic actor story (AS.dyn) by usage of real learning actor models (AM.learn)

Figure **??** shows the main outline of an actor story with really learning actors. The main points are the following ones: Every actor A of the population of participating actors has at a certain moment of time t

1. some *perception* π of the actual environment E,
2. some already gained *knowledge/ experience* K.
3. some *preferences (PRF)* what is more preferable in case of more than one option,
4. some possible *actions* ρ,
5. some *learning function* ϕ to compute possible changes of $\{K, PRF, \rho\}$.

Furthermore it is assumed that the following holds: The participating actors can

6. *interact* with each other either by
7. *communicating* with each other by using some *language (L)* or by
8. *coordinating* their behavior based on the communication.

TAMING DYNAMIC ACTORS: From these assumptions it follows that a *precise forecast* of all possible changes in an unrestricted dynamic actor story is not any more possible.

If one – by some reasons – is in need for a *certain course* of the actor story which can be repeated within given limits $+/-\epsilon$ then one has to embed the learning actors into some *training processes* τ where they will become trained to react in the states of the actor story with *prescribed responses* ρ. The prescribed responses should *match* the *prescribed actor story (AS.pre)* within some variance of $+/-\epsilon$.

In our *everyday world* there is a great demand of processes which fit expectations. This means that they widely are following predefined patterns. Thus there is usually no a great demand of dynamic processes. This regulates the needs for really learning artificial intelligent systems strongly.

One *side effect* of this strong bias for predefined processes is that the learning potential of real learning systems as animals and the homo sapiens is usually not exploited too much. In our everyday world *creative learning behavior* is mostly perceived as *dangerous* and un-productive.

7
AS - AM Interaction, Example 1

EXAMPLE 1 FOR AS-AM INTERACTION: This text describes a simple example for an *actor story - actor model interaction*. It takes the example of a simple actor story from chapter 4. In that example **??** there is a person as a user and an electronic door. The person is the *executive actor (eA)* and the electronic door is the *assistive actor (aA)*. While the executive actor can be assumed as a *real learning system eA.ls* the electronic door can be assumed as *non-learning* system which hints to a *deterministic system (aA.det)*.

DESCRIPTION OF AMs: Analogously to the different modes to describe an actor story one can for the description of actor models use different modes of description. In this simple example the following options will be used:

1. *TEXTUAL AM:* To begin the description of an actor model one can start with some *everyday language (L)* with its weaknesses but also strengthnesses.

2. *MATHEMATICAL MAM:* One can then translate the everyday version by some *mathematical expressions (L.math)*.

3. *PROGRAMMING AAM:* Finally if one wants to implement the actor model one can translate the mathematical version further into some *programming language (L.algo)*.

TEXTUAL AM (TAM): Here a simple text representing a textual version of both actor models.

TAM for the eA.ls:

$$\pi \quad : \quad \text{Ther is a visual perception of the environment} \tag{7.1}$$
$$K \quad : \quad \text{It is known to enter a code C to open the door} \tag{7.2}$$
$$PRF \quad : \quad \text{Use the code C} \tag{7.3}$$
$$\rho \quad : \quad \text{Press the keys of the keypad} \tag{7.4}$$
$$\phi \quad : \quad \text{Keep the pattern stable} \tag{7.5}$$
$$COM \quad : \quad \text{No further communication necessary} \tag{7.6}$$

TAM for the aA.det:

$$\pi \quad : \quad \text{Ther is a perception of the keys pressed} \tag{7.7}$$
$$K \quad : \quad \text{It is known to open the door after receiving code C} \tag{7.8}$$
$$PRF \quad : \quad \text{Stay with code C} \tag{7.9}$$
$$\rho \quad : \quad \text{Open the door if code C has been entered correctly} \tag{7.10}$$
$$\phi \quad : \quad \text{Follow the pattern} \tag{7.11}$$
$$COM \quad : \quad \text{No further communication necessary} \tag{7.12}$$

INTERACTIONS: Given the above TAMs one can describe the interactions of the actor story with regard to these actor models as follows:

1. *AS*: There is an electronic door D with a keypad K. The door is closed. Before the door stands a person A, which is able to enter a code C into the keypad K. *AM eA.ls:* The person sees the electronic door with its keypad closed, and while the person knows that it has to enter the code C to open the door the person decides to start the action to enter the code C. *AM aA.det:* The electronic door does not sense any key pressed therefore it stays unchanged.

2. *AS*: The person A enters a code C into the keypad K and causes an effect. *AM eA.ls:* The person pushes those keys of the keypad which correspond to the known code C. *AM aA.det:* The electronic door senses certain keys pushed. These correspond to its known code C. Therefore the electronic door opens the door.

3. *AS*: The door is open. The goal-state has been reached. *AM eA.ls:* The person perceives visually that the door has opened. The goal has been reached. *AM aA.det:* The electronic door has opened the door and stays quiet.

This approach is very informal. One still very excellent example how to formalize an actor model in accordance with empirical psychology is still the book of Card, Moran, and Newell (1983)[1]. They applied the whole apparatus of empirical psychology to the case of actors and their behavior. For the formalization they introduced in their chapter 5 additionally the GOMS model, which is in use until today.

[1] Stuart K. Card, Thomas P. Moran, and Allen Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah (NJ), 1 edition, 1983

8
Testing An AS

If an actor story AS has been constructed one has to check the *cognitive plausibility* of the actor story as well as the *usability* of the intended assistive actors (aAs) by the intended users.

The *cognitive plausibility* is located in the relationship between the *knowledge of the stakeholder* and the possible *experience* when *testing the actor story in a simulation*. If the real experience within a simulation *differs* from the given experience in the brains of the stakeholders than the cognitive plausibility of the actor story is low, eventually too low.

The *usability* of the intended assistive actors (aAs) is located in the relationship between the intended executive actors (eA) and a preliminary mock-up of the intended assistive actors (aA). While the intended executive actor tries to realize a process which is in agreement with the actor story it has to be *empirically measured* (i) to which degree the intended executive actors are *able to realize the actor story* with *this mock-up* and (ii) it should be *subjectively measured* to which degree the intended executive actor is *satisfied* with this process *in an emotional dimension*.

SIMULATION: Having an actor story AS and an *assisting simulator software* σ one can realize a *simulation*, either (i) purely *passive* without interactions or (ii) with *interactions*. In the case of an *interactive simulation* real actors can interact with the simulation and thereby *influence the course of the simulation.* A simulation enables a *shared experience* with a *common understanding* in all participants of the simulation. The *simulation experience* can be compared with the available *real-world experience* of the participants and this allows a special kind of a *cognitive test* revealing those aspects of the simulation which *differ* from the known reality. These *experienced differences* can shed some light on either *deficiencies* of the *simulation* or deficiencies of the *real world situation*.

The introduction of actor models (AMs) simultaneously to an actor story (AS) does not change the concept of a simulation. Actor models occur in the format of a change-rule which in turn is connected to an algorithm which defines its computations.

GAMING: If one extends an interactive simulation with the definition of explicit *win-lose states* then one can turn a simulation into a *game* with real actors which can compete and where some of the participant can become *winners*. Compared to simulations with their somehow *infinite* possibilities identifies a game in advance some *special states of interest* which narrows the scope of the analysis. This helps to focus the test of the process to these special states of interest and enables a much *faster clarification of research questions*. In this sense is *gaming* the *more efficient way of learning* by simulation.

VERIFICATION OF NFRs; ORACLE: If one has defined some NFRs (non-functional requirements) for the actor story then one can after the completion of an actor story including simulation *verify* whether the NFRs are *true* in the actor story with regard to the assumed environment *or not*. A special case of the verification of NFRs is the *oracle* function. Because the verification of NFRs is done in the manner of an *automated prove* with regard to the existence or non-existence of some defined property (associated with a NFR), one can use this mechanism also for to *check* whether a *special state of interest* will *occur or not occur* within a *defined time window* of *all* possible simulations. Such a mechanism can be of great help for the analysis of the possible future of a process, especially without having the need to do *all* the possible (interactive) simulations which is practically impossible on account of the needed time. But because such an oracle-process can only work with the given change-rules *as if these will not change* and without the *non-deterministic behavior of real executive actors* the oracle-results have to be used with caution.

NEED FOR MOCK-UPs: Until that point there exist only *symbolic descriptions* about possible real states. To turn the symbolic descriptions into a *real working system* one has to implement these descriptions into a real system. But such a full *implementation* is not the job of the AAI analysis. The AAI analysis only examines possible states and possible behavior profiles and checks with the aid of mock-ups whether these ideas will work sufficiently well. *Mock-ups* are physical systems which show all the main physical properties of the intended system without being a full implementation of this system.

USABILITY TESTING: *Usability* reveals something about the way how good the interaction of the intended executive actors with the intended assistive actor works within the whole actor story. Some of the questions which shall be answered by an usability test are: Is it too difficult for the executing actor to *learn* the needed behavior? Does the executing actor need *too much time*? Do continuously occur *too many errors*? To answer these and similar questions one has to prepare a *test scenario* which allows a real executing actor to behave according to the actor story by using the intended assistive actor realized as a *mock-up*. This test has to be managed by a *test coordinator* assisted by some *observing persons* or/ and *recording devices* to produce a *protocol* of the events during the test. The protocols have then to be converted into *test data* which can be used for analytical purposes.

A special point in the AAI usability testing is that within the AAI framework it is generally assumed that the executive actors are by default *learning systems*(which holds for all biological systems). This means that the executive actors eA all have an individual *behavior function* ϕ. This induces within a testing procedure the possible effects that the behavior of a executing actor can change from test to test.[1] To restrict the usability test therefore to only one test run is highly dangerous. It is recommended to *repeat an usability test at least three times.* What number n has to be assumed to be the *optimal* number is still an unanswered question.

[1] Which is indeed the normal case. Therefore you can find in all reports about learning experiments always so-called *learning curves* representing these changes along a time line.

Part II

Application

Bibliography

Claude Audoib and Bernard Guinot. *The Measurement of Time. Time, Frequency and the Atomic Clock*. Cambridge University Press, Cambridge - New York - Melbourne, 1st edition, 2000. The original French edition of the book appeared 1998.

W. Balzer, C. U. Moulines, and J. D. Sneed. *An Architectonic for Science*. D.Reidel Publishing Company, Dordrecht (NL), 1 edition, 1987.

N. Bourbaki. *Éléments de Mathématique. Théorie des Ensembles*. Hermann, Paris, 1 edition, 1970.

Stuart K. Card, Thomas P. Moran, and Allen Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah (NJ), 1 edition, 1983.

G. Doeben-Henisch and M. Wagner. Validation within safety critical systems engineering from a computational semiotics point of view. *Proceedings of the IEEE Africon2007 Conference*, pages Pages: 1 – 7, 2007. DOI: 10.1109/AFRICON.2007.4401588.

Gerd Doeben-Henisch. Semiotic Machines. In *Signs and Space - Raum und Zeichen. An International Conference on the Semiotics of Space and Culture*, pages 313–327, Tübingen (DE), 1998. Gunter Narr Verlag.

Gerd Doeben-Henisch. Reconstructing human intelligence within computational sciences: An introductory essay. In A. Loula, R. Gudwin, and J. Queiroz, editors, *Artificial Cognition Systems*, pages 106–139, Hershey - London - Melbourne - Singapore, 2007. Idea Group Publishing.

Louwrence Erasmus and Gerd Doeben-Henisch. A theory of the system engineering process. In *9th IEEE AFRICON Conference*. IEEE, 2011a.

Louwrence Erasmus and Gerd Doeben-Henisch. A theory of the system engineering management processes. In *ISEM 2011 International Conference*. ISEM, 2011b. Conference 2011, September 21-23, Stellenbosch, South Africa.

Jonathan Grudin. A Moving Target: The Evolution of HCI. In A. Sears and J.A. Jacko, editors, *The Human-Computer Interaction Handbook. Fundamentals, Evolving Technologies, and emerging Applications*. 2 edition, 2008.

Yuval Noah Harari. *21 Lessons for the 21st Century*. Spiegel & Grau, Penguin Random House, New York, 2018.

Charles Morris. *General Theory of Signs*. Mouton, Paris, 1971.

Charles W. Morris. Foundations of the Theory of Signs. volume 1 of *International encyclopedia of unified science*, pages 1–59. The University of Chicago Press., Chicago (Illinois), 1938. ISBN 0226575772.

Winfried Nöth. *Handbook of Semiotics*. Indiana University Press, Bolomington - Indianapolis, 1 edition, 1990. Enlarged and completely rewritten edition of the 'Handbuch der Semiotik' (1985).

Winfried Nöth. *Handbuch der Semiotik*. J.B.Metzler, Stuttgart - Weimar, 2nd edition, 2000. Completely rewritten 2nd edition of the 'Handbook of Semiotics' (1990).

Richard W. Pew. Introduction. Evolution of human-computer interaction: From memex to bluetooth and beyond. In J.A. Jacko and A. Sears, editors, *The Human-Computer Interaction Handbook. Fundamentals, Evolving Technologies, and emerging Applications*. 1 edition, 2003.

Eric Schmidt and Jared Cohen. *The New Digital Age. Reshaping the Future of People, Nations and Business*. John Murray, London (UK), 1 edition, 2013. URL https://www.google.com/search?client=ubuntu&channel=fs&q=eric+schmidt+the+new+digital+age+pdf&ie=utf-8&oe=utf-8.

J. D. Sneed. *The Logical Structure of Mathematical Physics*. D.Reidel Publishing Company, Dordrecht - Boston - London, 2 edition, 1979.

G.J. Whitrode. *The Natural Philosophy of Time*. Clarendon Press, Oxford, 2nd edition, 1980. The 1st edition of the book appeared 1961.

Index