PART I: INTRODUCTION

Have you ever thought about the complexity of the world in which we live and about the complexity of ourselves? And this world is becoming even more complex with the rapid development of information technology. Millions or billions of information processing stages are going on in our brain before we perform an action. Thousands of operations and many human and technical agents are involved while you for instance purchase a pizza via Internet or while you are travelling by plane. Understanding behavior of complex systems, for example complex socio-technical systems where humans interact with computers, is not a trivial task while it is crucial for improving human well-being and quality of life and work conditions. We need to understand socio-technical systems in order to predict and control their behavior and in order to design better systems in future. The current research has an aim to analyze the complexity of human behavior at different levels in two socio-technical domains: the *computerized health behavior support domain* such as computerized therapy, mobile lifestyle support and *safety critical domains* such as air traffic management and naval warfare. The former represents an example of human functioning in private life and the latter is an example of human functioning in critical and highly dynamic work environments. Both domains address human functioning in abnormal contexts: in the case of healthcare it concerns the life of chronic patients and in the case of safety critical domains it involves functioning of humans in highly demanding and dynamic circumstances.

Complex Systems Analysis

Analysis of complex systems is performed in different disciplines, using different methods and at different levels. Traditionally, complex systems analysis was performed in biology, physics and chemistry, later with the development of distributed computer systems, the complexity analysis paradigm became popular in computer systems engineering. Traditional methods of dealing with complex systems can be described as analytical: by decomposing a system into its parts and by the analysis of the parts we try to infer the properties of the whole system [10]. This analytic approach is based on the principle of reducibility, or sometimes even reductionism. It is well understood that reductionism has its limitations: for example knowing the chemistry does not mean that we understand life. Another approach of dealing with complex systems is analysing them at a global, or macroscopic level. For instance we do not describe a gas by providing all individual coordinates of its atoms at each time point, but rather in terms of macroscopic quantities such as pressure and temperature; or we do not describe human behaviour in terms of activation of particular types of neurons in given brain regions, but in terms of actions that are being performed. The challenge of relating these two levels of descriptions still remains open in all scientific fields.

Nowadays new interdisciplinary studies have emerged that try to bring together and to summarize complex systems analysis from different fields. For example, the *complex systems* discipline proposes different computational methods and approaches towards analysis of complex systems, such as network theory, general systems theory, agent-based modelling. One of the promising approaches for the analysis of complex systems is the *system of systems* (SoS) science that focuses on hierarchical organisation of elements in a system in order to explain its complexity. The theoretically grounded idea of system of systems was first introduced by Simon [15]. The author adopted it from biological organisations. The main idea of this approach is that any complex organisation can be split into different hierarchical levels similar to biological systems: e.g. functioning of an organism is contingent on functioning of organs, each organ is in its turn dependent on the processes at its cellular level and the cells can be split into more microscopic levels of organic substances etc. This hierarchical paradigm was adopted by the author as a theoretical framework for the design and analysis of complex systems. DeLaurentis [1] applied it in his framework of organizational architecture for addressing the equity issue in multi-actor policymaking concerning aviation's emissions reduction. He demonstrated how the formal SoS approach can provide an insight into the relations between the actor in an organization and how it can inform policymaking in determining the magnitude of compensation required when a particular solution is pursued.

From this hierarchical perspective, behaviour of the whole system at a global level is contingent on its micro level elements and their interactions. However, it is not always clear how the global behaviour emerges from local parts. Socio-technical systems such as air traffic management, modern warfare or eHealth can be regarded as 'complex' systems since human behaviour that is complex in itself is intermingled with the behaviour of numerous technical systems. In this blending the nature of the whole system might not be determined and predicted by the analysis of some parts and such systems exhibit emergent behaviour.

Interlevel Relations in Complex Systems

Complex systems can be described and analysed at different levels: starting from very low levels, such as biological cells, atoms and even electrons, to higher levels, such as living organisms, populations, mechanical units and machines. However, the problem of connectivity between different types of descriptions and the challenge of bridging the levels still exist. A classical mind-body problem can be an example of this challenge. For instance, properties of human brain defined at a neurological level impose constraints on mental properties and human behaviour, though not every mental property is associated with the same neurological property; thus the determination between mind and brain is asymetrical and it is not trivial to relate neurological and psychological levels of description [2]. Models of complex systems at different levels of description demonstrate the same challenge: how can we relate properties of models at a global level to the properties that characterize low-level models?
A three-dimensional abstraction framework for modelling and analysis of complex systems was proposed in [7]. This framework allows for a direct and transparent specification of a system's level of description used in a particular model with subsequent possibility of establishing interlevel relations between the models with different levels of representation. By means of direct mapping between models at different levels more insight can be obtained in the phenomenon of emergence in complex systems. According to this

framework, one of the important choices made in (multi)agent system modelling and analysis in a certain application area is the grain-size or abstraction level of the model being developed. Models at a global level are too coarse-grained (too abstract) and they may miss details of the domain modelled. The details could be essential to the aim of the model. On the other hand, models which are too fine-grained (too detailed) for the aim of the model may become difficult to handle because they are too complex and/or not transparent. Therefore choosing the right level of abstraction may be crucial. However, the notion of abstraction level itself is not sufficient according to the authors [7]. For example, the following questions arise in this respect: 'Does more abstract mean that the models for the internal processes within agents are described at a more abstract level (abstracting from physiological, cognitive and affective detail)? Or should it be interpreted such that the models describe relationships over larger time intervals (abstracting from the smaller time steps)? Or does more abstract mean that within the multi-agent system model higher level structures are used that aggregate individual agents (considering groups or clusters of agents as entities, as in organisation models)?' [7]. This framework clarifies the notion of abstraction level and claims that the point of departure should be the consideration of different dimensions of abstraction levels. The following dimensions of abstraction are proposed: the *process abstraction,* the *temporal,* and the *agent cluster* dimension (see Fig. 1). For each of these dimensions a (multi)agent system application can be modelled from a more local level perspective and from a more global-level perspective. Thus a framework is obtained to distinguish abstraction levels for different types of models. The authors demonstrate that this framework is able to distinguish and to position different types of multi-agent system models known from the literature on multi-agent systems, including, for example, population-based vs. agent-based models, behavioural vs. cognitive agent models, and executable models vs. requirements models. The distinctions made by the framework are purely semantic, and independent of any representation format of a model.

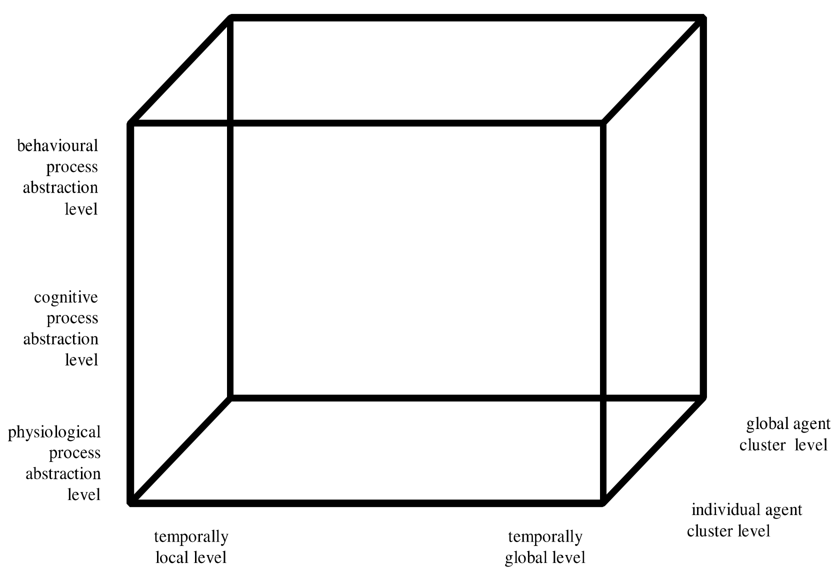

Figure 1 (adopted from [7]). Three-dimensional classification framework for modelling and analysis of (multi-agent) systems.

In spite of the fact that the framework was developed with multi-agent systems in mind, it can be applied to any complex organisation model where multiple actors are involved. The proposed classification incorporates the System of Systems (SoS) hierarchical framework described in the previous subsection: the concept of the grain size emphasised by the authors reflects the hierarchical abstraction framework from the SoS, but is defined in a continuous manner.

Research described in this thesis has been performed in the context of this framework.

Research objectives

Both safety critical domains, such as air traffic management, warfare, and computerized health support, such as computerized therapy or mobile lifestyle support, are the examples of systems that have high social relevance and are defined by complex interrelationships within them. Air traffic management is a complex socio-technical system that is characterized by high dynamics, intricate interactions between multiple agents and that exhibits emergent behavior. For instance, one can analyze the character and the nature of individual components of a system in order to understand its principal behavior, e.g. one can understand why an air traffic accident occurred and which agents were involved in the accident. However, it is not trivial to predict new global behavior of the system given the behavior of local components: one may not realize why an accident may occur in the future for this particular system in a particular context even if the behavior of individual agents, or components, is predefined.
Another example of a system where emergent behavior is observed is computerized health behavior support. In the context of the lifestyle domain, in spite of the fact that an individual is aware of the necessity of a healthy lifestyle and his attitude to behavior change is positive, it is quite hard to adopt new behavior, and therefore hard to predict when the individual will actually change his or her behavior. Moreover, it is difficult to predict the interactions of humans with technical systems in the computerized health behavior support domain since this is a relatively new phenomenon that has not been extensively studied in a systematic manner yet. The work described in this thesis focuses on these two domains and addresses the analysis of socio-technical systems by means of formal methods.

The aim of the current research is twofold:

(1) to explore the possibilities of agent-based modelling in the analysis of complex socio-technical systems and the interactive health coaching systems

(2) to provide an adequate support and advice based on this analysis, either direct at the level of an individual or indirect in the form of general recommendations concerning the system's design.

The complex systems analysis tries to relate different levels of a system's description and to understand global emergent behavior of a whole system based on the local behavior of the system's constituents using agent-based modelling.
The first research objective (1) can be subdivided in the following sub-questions:

a) How can global behavior of a socio-technical system be explained by the behavior and interactions of its local components (top-down system analysis)?

b) How does a change in the behavior of a local component result in global behavior of the system (bottom-up system analysis)?

c) How can different modelling approaches and techniques to study socio-technical systems be compared and contrasted with respect to analysis possibilities and complementarity?

The second research objective (2) can be subdivided into the following sub-questions:

d) How can computational modelling and analysis at different levels contribute to the development of ambient support systems in computerized health behavior support and safety critical domains?

e) How can computational modelling and analysis at different levels contribute to the development of recommendations in computerized health behavior support and safety critical domains?

An overview of the contributions of each chapter of the thesis related to the above listed research sub-questions is given in Table 1.
For both safety critical domains and computerized health behavior support, models of human functioning were created and analyzed with the aim to provide support and recommendations at the level of the whole system and at the level of parts of the system. For instance, Part II of the thesis is dedicated mainly to the analysis of safety-critical systems and Part III is focused on both analysis and support of humans with respect to their health-related behavior within the intelligent health support context.

Table 1. Answered research questions per chapter.

Research questions	Research subquestions	Chapters
(1) How can agent-based modelling contribute to the analysis of complex socio-technical systems?	a) How can a global behavior of a socio-technical system be explained by the behavior and interactions of its local components (top-down system analysis)?	2, 4, 10
	b) How does a change in the behavior of a local component result in global behavior of the system (bottom-up system analysis)?	1, 2, 4, 5, 7, 8, 9, 10
	c) How can different modelling approaches and techniques to study socio-technical systems be compared and contrasted with respect to analysis possibilities and	3, 6

	complementarity?	
(2) How can agent-based analysis of complex socio-technical systems help in providing an adequate support for humans in abnormal contexts?	d) How can computational modelling and analysis at different levels contribute to the development of ambient support systems in eHealth and safety critical domains?	1, 7, 8, 9, 10
	e) How can computational modelling and analysis at different levels contribute to the development of useful recommendations in eHealth and safety critical domains?	2, 3, 4, 5, 6, 9

Research methodology

The general methodology adopted in the thesis consists of three main steps:

1. Modelling a system's behavior
2. Analysis of a system's behavior at different levels of abstraction according to the framework of interlevel relations described in [7]
3. Generation of support or recommendations based on the system's analysis

During step 1 a computational model is developed. This step consists of *conceptualization* and *formalization* phases. Conceptualization refers to the selection and determination of the main system's components and the relations between them. The components are defined according to the goals of a designer and are based on the knowledge obtained from the relevant literature, during the communication with domain experts or common sense knowledge. For instance, in order to model an air traffic incident in an airport, one should include the agents that were involved in the incident and the elements of an airport that played a role in the occurrence of the incident, define the agent's cognitive and behavioral components that are important for the incident. Further, during the formalization phase, the knowledge about the concepts is described in a formal language and is implemented with the help of particular software.

It is important to perform an analysis of a complex system according to a particular modelling framework. In this thesis an interlevel relations paradigm was applied to the analysis of human functioning and safety in critical domains and in eHealth and lifestyle. Namely, in both domains local, intermediate and global levels of analysis are distinguished across three attributes, or dimensions: *process abstraction*, *agent cluster* and *time* [7].Thus, Step 2 comprises performing analysis of the system at different levels of abstraction by experimenting with the model: performing simulations, generating simulation traces and applying formal analysis to the simulation traces or model-based reasoning both at a local and at a global level of the system's functioning. The interlevel analysis is performed across one of the three dimensions in the scope of the three-dimensional models' classification framework.

Based on the results of the analysis performed during step 2, during step 3 direct context-dependent support is provided to humans, either as part of an autonomous support system or in the form of general recommendations and implications on system's design and functioning.

Modelling approach

In this subsection the general modelling approach and modelling languages that are applied during the research are described.

Agent-based modelling

Agent-oriented approaches have been widely used for modelling complex socio-technical systems [13]. The essence of agent-based modelling is the agent-oriented world view that implies that the world consists of active, purposeful agents that interact to achieve their objectives. There is still much debate, however, about what exactly constitutes an agent or agenthood. The majority of researchers agree on the following definition of an agent proposed in [17]: 'An agent is an encapsulated (computer) system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives'. The agent-based approach for modelling is characterized by flexibility of outcomes and the strong probability of the system's emergent behavior that cannot be predicted from behavior of its individual components, or agents. This is particular relevant for modelling human cognition and complex-socio-technical systems where the interactions between elements are not trivial and where systems exhibit emergent behavior.
The class of agent models that have beliefs (what the agent knows about the world and other agents), desires (what the agent wants or which goals he has) and intentions (what the agent intends to do) have been used in a wide variety of applications including air traffic control, process control and transportation [13]. This Belief-Desire-Intention (BDI) paradigm is powerful for the representation of cognitive states of agents and has been adopted in (part of) the current work in order to model internal cognitive states of human agents that may be crucial for understanding of human health related behavior in the context of eHealth and human performance in safety critical domains and in air traffic in particular.

Modelling framework

For the overall architecture of the agent models in this thesis, principles of component-based agent design have been followed, as, for example, are described in [8] within the agent design method DESIRE. This architecture is characterized by structural decomposition of an agent's elements and is designed as a General Agent Model (GAM): see [5], [6] and the classification of agents' architectures in [9]. According to this architecture, knowledge about the world in the form of a *domain model* is needed in order to create an ambient agent model or model of a human agent in a socio-technical system. Further, two main components are distinguished within the ambient agent model at a lower level: the *analysis* component and the *support* component (see Figure 2). Accordingly, two different ways to integrate the domain models within the agent model have been used; see Figure 2.

- *analysis component*

This component is needed in order to perform analysis of the human's states and processes by (model-based) reasoning based on observations and the domain model.

- *support component*

This component plays a role of generating support actions for the human by (model-based) reasoning based on observations and the domain model.

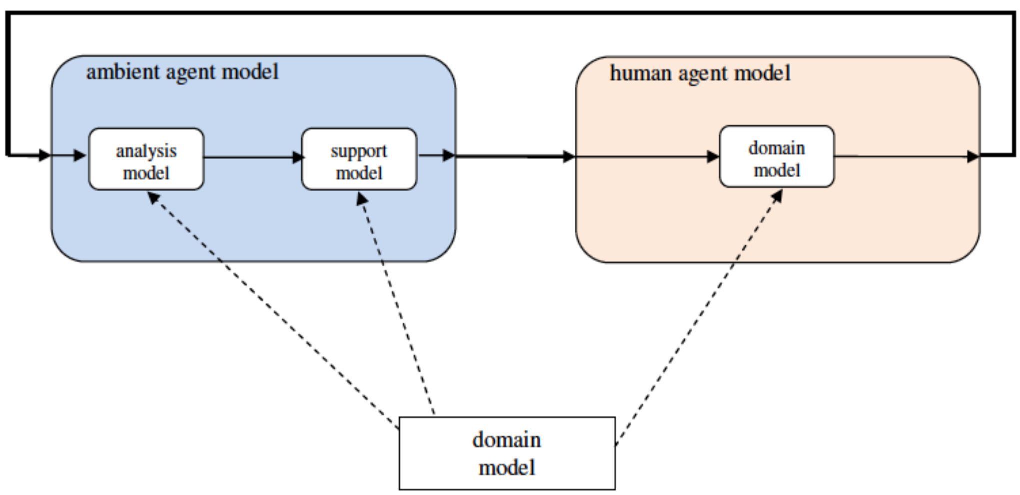

Figure 2 (adopted from [5]). Overall design of the ambient agent and the integration of the domain model. Here solid arrows indicate information exchange (data flow) and dotted arrows the integration of the domain model within the agent model.

In Part II of the thesis the research is predominantly focused on the development and analysis of domain models of human functioning with implications for future ambient support while in Part III ambient agent analysis and support models development is emphasized.

Modelling Languages and Environments

Most agent-based models described in this thesis were implemented in the LEADSTO [5] or Matlab [19] environment. Formal analysis of models in Chapters 2, 3 and Chapter 8 was performed in TTL [5]. The analysis of the model in Chapter 3 was made with the help of the Delphi [18] and Matlab software environments, the models in Chapters 5,7, 8, 9, 10 were implemented in Matlab and the model's empirical validation in Chapter 8 was executed with the help of the statistical package SPSS [20].

TTL and TTL Checker

The predicate-logical Temporal Trace Language (TTL) described in [4] integrates qualitative, logical aspects and quantitative, numerical aspects. This integration allows the modeler to exploit both logical and numerical methods for analysis and simulation. It can be used to express dynamic properties at different levels of aggregation, which makes it well suited both for simulation and logical analysis.
The TTL language is based on the assumption that dynamics can be described as an evolution of states over time. The notion of state as used here is characterized on the basis of an ontology defining a set of physical and/or mental (state) properties that do or do not hold at a certain point in time. These properties are often called *state properties* to distinguish them from dynamic properties that relate different states over time. A specific state is characterized by dividing the set of state properties into those that hold, and those that do not hold in the state. Real value assignments to variables are also considered as possible state property descriptions.
To formalise state properties, ontologies are specified in a (many-sorted) first order logical format: an *ontology* is specified as a finite set of sorts, constants within these sorts, and relations and functions over these sorts (sometimes also called signatures). The examples mentioned above then can be formalised by n-ary predicates (or proposition symbols), such as, `moves_with_velocity(A, S)` or `communicate_from_to(C, A, information_element)`. Such predicates are called *state ground atoms* (or *atomic state properties*). For a given ontology `Ont`, the propositional language signature consisting of all ground atoms based on `Ont` is denoted by `APROP(Ont)`. One step further, the *state properties* based on ontology `Ont` are formalised by the propositions that can be made (using conjunction, negation, disjunction, implication) from the ground atoms. Thus, an example of a formalised state property is `moves_with_velocity(A, S)` & `communicate_from_to(C, A, information_element)`. Moreover, a *state* S is an indication of which atomic state properties are true and which are false, i.e., a mapping S: `APROP(Ont)` → `{true, false}`. The set of all possible states for ontology `Ont` is denoted by `STATES(Ont)`.
To describe dynamic properties of complex processes such as in aviation, explicit reference is made to *time* and to *traces*. A fixed time frame T is assumed which is linearly ordered. Depending on the application, it may be dense (e.g., the real numbers) or discrete (e.g., the set of integers or natural numbers or a finite initial segment of the natural numbers). Dynamic properties can be formulated that relate a state at one point in time to a state at

another point in time. A simple example is the following (informally stated) dynamic property about the absence of collisions:

For all traces γ,
there is no time point t
on which a collision takes place.

A *trace* γ over an ontology `Ont` and time frame T is a mapping γ : T → `STATES(Ont)`, i.e., a sequence of states γ_t ($t \in T$) in `STATES(Ont)`. The temporal trace language TTL is built on atoms referring to, e.g., traces, time and state properties. For example, 'in trace γ at time t property p holds' is formalised by `state(γ, t) |= p`. Here |= is a predicate symbol in the language, usually used in infix notation, which is comparable to the Holds-predicate in situation calculus. *Dynamic properties* are expressed by temporal statements built using the usual first-order logical connectives (such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$) and quantification ($\forall$ and $\exists$; for example, over traces, time and state properties). For example, the informally stated dynamic property introduced above is formally expressed as follows:

```
∀γ:TRACES ¬∃t:TIME
state(γ, t) |= collision
```

The TTL Checker tool based on the TTL language [2] takes as input a formula expressed in TTL and a set of traces, and verifies automatically whether the formula holds for the traces.

LEADSTO

The executable LEADSTO language and software environment described in [4] has been defined as a sublanguage of TTL, with the specific purpose to develop simulation models in a declarative manner. In LEADSTO, direct temporal dependencies between two state properties in successive states are modelled by *executable dynamic properties.* The LEADSTO format is defined as follows. Let α and β be state properties as defined above. Then, $\alpha \twoheadrightarrow_{e, f, g, h} \beta$ means:

If state property α holds for a certain time interval with duration g,
then after some delay between e and f
state property β will hold for a certain time interval with duration h.

The LEADSTO Simulation Environment [4] takes a specification of executable dynamic properties as input, and uses this to generate simulation traces.

Matlab

MATLAB® is a high-level language and interactive environment for numerical computation, visualization, and programming developed by MathWorks [19]. MATLAB is being widely used to analyze data, develop algorithms, and create models and applications.

MATLAB allows for high computational efficiency, even with large amounts of data. The language, tools, and built-in math functions enable to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages. MATLAB made it possible to effectively visualize the results of simulations and to run many simulations in a relatively short amount of time.

IBM SPSS

The software package SPSS was originally developed for statistical analysis in social sciences and became among the most widely used programs for statistical analysis in that field [20]. Nowadays it is used by market researchers, health researchers, survey companies, government, education researchers, marketing organizations, and others. In addition to statistical analysis, data management (case selection, file reshaping, creating derived data) and data documentation (a metadata dictionary is stored in the datafile) are features of the base software.
In this thesis, IBM SPSS is used for statistical analysis of empirical data, such as bivariate Pearson Product-Moment correlation analysis of results obtained from subjects' questionnaires in Chapter 8.

Delphi

Delphi is an object-oriented programming language that was developed by the software company Borland in 1994 [18]. It contains a set of extensions to standard Pascal. Delphi Pascal is a high-level, compiled, strongly typed language that supports structured and object-oriented design. It is characterized by an easy-to-read code, quick compilation, and the use of multiple unit files for modular programming. Selected events in a dynamic risk agent-based model in Chapter 4 were programmed in Delphi.

Thesis outline

This thesis is based on a collection of articles, which contain some overlapping information concerning the modelling context and methodologies. Each article represents a chapter in the thesis and can be read independently. The thesis consists of four main parts: Introduction, Safety Analysis in Critical Domains, Healthcare and Lifestyle and Discussion. Below, each part of the thesis is outlined.

Part I: Introduction

In this part the context and the scope of the research are given, research objectives and research questions are introduced, the general methodology and modelling approaches and techniques are described and an overview of the structure and the content of the thesis is given. The thesis focuses on agent-based modelling and analysis of complex socio-technical systems at different levels of abstraction. Examples of such systems are analyzed in two domains: safety critical domains and healthcare. The main research question within this thesis has an aim to understand a global emergent behavior of a whole system based on the local behavior of the system's constituents and to provide adequate support and advice, either direct at the level of an individual or indirect in the form of general recommendations concerning the system's design.

Part II: Safety Analysis in Critical Domains

This part is dedicated to the analysis and support of human functioning in safety critical and dynamic environments, such as air traffic management or warfare. Here models of an adaptive display and incident models based on some case studies are introduced and formally analyzed. Chapter 1 describes a model for information presentation developed for an adaptive display and its integration with an existing Operator Functional State model (OFS) [3]. An intelligent display introduced in this chapter is sensitive to an operator's functional state, such as stress, high workload etc. and adapts the presentation of information. The focus of this ambient system lies in the operator's comfort and safety. Chapter 2 describes the formal analysis of an aviation incident at different levels based on a real life case study by means of formalization and dynamic simulations. The case study describes an incident that occurred in one European airport in 1995. The incident analysis is performed at local, intermediate and global levels at the temporal and process abstraction dimension of the multi-agent system. Chapter 3 contrasts agent-based analysis of aviation accidents with other popular accident analysis approaches in aviation, such as Event Trees, FRAM and STAMP. In this chapter four approaches were applied to one case study and a comparative analysis of the approaches is given. In Chapter 4 a formal analysis of accident risk and conflict resolution events of a future TIPH (Taxiing into Position and Hold) air traffic operation is described. It was done by means of definition of relevant events in a stochastic dynamic risk model of the TIPH operation [16] and calculation events probabilities during Monte Carlo simulations of the model. Chapter 5 presents an integration of three agent-based models and an application of the integrated model to a real life incident analysis. The computational models that were integrated are Operator Functional State model (OFS) [3], Situation Awareness model [12] and decision-making. In Chapter 6 two different modelling approaches are contrasted: a population-based approach at an aggregated agents cluster level and an agent-based approach at an individual level. Both models were applied to the context of spread of Situation Awareness in a group.

Part III: Computerized Health Behaviour Support

In this part the possibilities of agent-based modelling and support are explored in the domain of health related behavior change. The scope of these types of models is quite broad: the models can be used both for chronic patients and for healthy individuals who want to change their health related behavior, to develop new healthy habits, to become aware of old habits or to adopt a healthy lifestyle. The models were developed at different levels of abstraction: both at an individual and at a group level across the agents clustering dimension; physiological, cognitive and behavioral across the process abstraction dimension; and at a local and global levels in time dimension. In Chapter 7 an intelligent support model for diabetic patients is described. The model gives advice about insulin intake based on the blood sugar level and daily activities and simulates the effect of advised insulin intake on the blood sugar level of an individual with diabetes type 2. Chapter 8 proposes a computational model of habit learning to support lifestyle change at an individual level of agents clustering dimension. In Chapter 9 a computational model of habit learning to support lifestyle change with social influence is presented. The model can be used in order to perform large scale agent-based simulations and can be incorporated in a personal ambient intelligent agent that can enable lifestyle support. In Chapter 10 an intelligent model-based multi-level support system for therapy adherence and behavior change and its preliminary evaluation is described.

Part IV: Discussion

In this part the discussion and conclusions of the current research are presented: the main contributions of the research, answers to the research questions, its limitations, lessons learned and some implications for the future work.

References

1. DeLaurentis, D.A. (2005). A Taxonomy-based Perspective for Systems of Systems Analysis
   Design Methods. IEEE International Conference on Systems, Man and Cybernetics, pp. 86-91.
2. Bem, S., & de Jong, H. L. (2006). Theoretical Issues in Psychology: An Introduction. SAGE.
3. Bosse, T., Both, F., van Lambalgen, R.M., and Treur, J. (2008). An Agent Model for a Human's Functional State and Performance. In: Proceedings of the Eighth IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT'08. IEEE Computer Society Press, vol. 2, pp. 302-307.
4. Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A. and Treur, J. (2009). Specification and Verification of Dynamics in Agent Models. International Journal of Cooperative Information Systems, volume 18, 2009, pp. 167-193.
5. Bosse, T., Hoogendoorn, M., Klein, M., van Lambalgen, R., van Maanen, P. P., & Treur, J. (2011). Incorporating human aspects in ambient intelligence and smart environments. Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives. IGI Global, pp. 128-164.
6. Bosse, T., Hoogendoorn, M., Klein, M. C., & Treur, J. (2010). A Generic Architecture for Human-Aware Ambient Computing. Agent-Based Ubiquitous Computing. Atlantis Press, 2010, pp. 35-61.
7. Bosse, T., Hoogendoorn, M., Klein, M.C.A., and Treur, J. (2010). A Three-Dimensional Abstraction Framework to Compare Multi-Agent System Models. In:

Pan, J.-S., Chen, S.-M., and Nguyen, N.T. (eds.), Computational Collective Intelligence: Technologies and Applications, Proceedings of the Second International Conference on Computational Collective Intelligence, ICCCI'10, Part I. Lecture Notes in Artificial Intelligence, vol. 6421, pp. 306-319. Springer Verlag, 2010.

8. Brazier, F.M.T., Jonker, C.M., and Treur, J., (2002). Principles of Component-Based Design of Intelligent Agents. Data and Knowledge Engineering, vol. 41, 2002, pp. 1-28.
9. Briot, J.-P., Meurisse, T., and Peschanski, F. (2007). Architectural design of component-based agents: A behavior-based approach. Programming Multi-Agent Systems. Springer Berlin Heidelberg, 2007, pp. 71-90.
10. Haken, H. (2006). Information and self-organization: a macroscopic approach to complex systems. Springer, 2006.
11. Hilburn, B. (2004). Cognitive complexity in air traffic control: A literature review. EUROCONTROL, Experimental Center. Bretigny-sur-Orge: EEC Note No. 04/05.
12. Hoogendoorn, M., van Lambalgen, R.M., and Treur, J. (2011). Modeling situation awareness in human-like agents using mental models. In: Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, IJCAI’11 - Vol. Two. AAAI Press, pp. 1697-1704.
13. Jennings, N.R. (2000). On agent-based software engineering. Artificial Intelligence 117, pp. 277-296.
14. Paries, J. (2011). Complexity, emergence, resilience…In Hollnagel, E., Paries, J., Woods, D.D. & Reathall, J., Resilience engineering in practice: A guidebook. England: Ashgate.
15. Simon, H. (1973). The organisations of complex systems. In H. Pattee, Hierarchy theory - The
challenge of complex systems. New York: George Braziller.
16. Stroeve, S.H., Doorn, van B.A., Bakker, G.J. (2012). Safety Assessment of a Future Taxi into Position and Hold Operation by Agent-Based Dynamic Risk Modelling. Journal of Aerospace Operations, Vol. 1, 2012, pp. 107-127.
17. Wooldridge, M. (1997). Agent-based software engineering, IEE Proc. Software Engineering 144 (1), pp. 26-37.
18. http://www.delphibasics.co.uk
19. http://www.mathworks.com
20. http://www-01.ibm.com/software/analytics/spss/