DEMSY – A SCENARIO FOR AN INTEGRATED DEMONSTRATOR IN A SMARTCITY

Constanze Deiters, Michael Köster, Sandra Lange, Sascha Lützel, Bassam Mokbel, Christopher Mumme, Dirk Niebuhr (Eds.)

7. Juni 2010

NTH Computer Science Report 2010/01

This work was funded by the NTH Focused Research School for IT Ecosystems. NTH (Niedersächsische Technische Hochschule) is a joint university consisting of Technische Universität Braunschweig, Technische Universität Clausthal, and Leibniz Universität Hannover.

niedersächsische technische hochschule · Technische Universität Braunschweig · TU Clausthal – Clausthal University of Technology · Leibniz Universität Hannover

IMPRESSUM

Publisher

NTH Focused Research School for IT Ecosystems
Technische Universität Clausthal, Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany

Editors of the series

Prof. Dr. Christian Müller-Schloer,
Prof. Dr. Andreas Rausch,
Prof. Dr. Lars Wolf

Technical editor

Dirk Niebuhr
Contact: dirk.niebuhr@tu-clausthal.de

ISBN 978-3-942216-13-5

AUTHORS

Prof. Dr. Jiří Adámek	TU Braunschweig
Prof. Dr.-Ing. Michael Beigl	TU Braunschweig
Dipl.-Inf. Martin Berchtold	TU Braunschweig
M. A. Kerstin Bischoff	Leibniz Universität Hannover
Dipl.-Math. Olesia Brill	Leibniz Universität Hannover
Dipl.-Inf. Constanze Deiters	TU Clausthal
Prof. Dr. Jürgen Dix	TU Clausthal
Dipl.-Ing. Aret Duraslan	Leibniz Universität Hannover
M. Sc. Gianina Gănceanu	TU Braunschweig
Prof. Dr. Ursula Goltz	TU Braunschweig
Prof. Dr. Barbara Hammer	TU Clausthal
Dr. Michaela Huhn	TU Braunschweig
Dipl.-Ing. Tereza Iofciu	Leibniz Universität Hannover
Dipl.-Inf. Christoph Knieke	TU Braunschweig
Dipl.-Inf. Michael Köster	TU Clausthal
Dipl.-Inf. Sandra Lange	TU Clausthal
Dipl.-Math. Peter Lohmann	Leibniz Universität Hannover
Dipl.-Inf. Sascha Lützel	TU Clausthal
B. Sc. Benjamin Mensing	TU Braunschweig
Dipl.-Inf. Bassam Mokbel	TU Clausthal
Prof. Dr. Jörg P. Müller	TU Clausthal
Prof. Dr. Christian Müller-Schloer	Leibniz Universität Hannover
Dipl.-Wirt.-Inf. Christopher Mumme	TU Clausthal
Prof. Dr. Wolfgang Nejdl	Leibniz Universität Hannover
Dipl.-Inf. Dirk Niebuhr	TU Clausthal
Prof. Dr. Andreas Rausch	TU Clausthal
Dipl.-Ök. Marc-Oliver Reeh	Leibniz Universität Hannover
Dipl.-Psych. Klaus Reinprecht	TU Braunschweig
M. Sc. Sebastian Schildt	TU Braunschweig
Prof. Dr. Kurt Schneider	Leibniz Universität Hannover
Dipl.-Math. Christian Schulz	Leibniz Universität Hannover
Prof. Dr. Christian Siemers	TU Clausthal
Dipl.-Ök. Dieter Varelmann	Leibniz Universität Hannover
Prof. Dr. Heribert Vollmer	Leibniz Universität Hannover
Prof. Dr. Mark Vollrath	TU Braunschweig
Prof. Dr. Bernardo Wagner	Leibniz Universität Hannover
Prof. Dr. rer. pol. Klaus-Peter Wiedmann	Leibniz Universität Hannover
Prof. Dr. Lars Wolf	TU Braunschweig

DemSy – A Scenario for an Integrated Demonstrator in a SmartCity

ABSTRACT

IT ecosystems – systems composed of a large number of distributed, decentralized, autonomous, interacting, cooperating, organically grown, heterogeneous, and continually evolving subsystems – are the future system generation. Today's state of the art does not enable us to develop these systems. We need to provide new methodologies to achieve dependable ***IT ecosystems***. Within the NTH Focused Research School for IT Ecosystems, we will provide such methodologies.

This technical report describes a fictional scenario where we accompany users at a smart airport. The smart airport itself can be seen as an ***IT ecosystem*** due to the complexity of the interacting systems present there. We describe one exemplary scenario how these users interact with the ***IT ecosystem***. Based on this scenario we motivate the research questions which will be addressed by different research projects of the NTH Focused Research School for IT Ecosystems.

In the future we will provide a joint demonstrator of an ***IT ecosystem***, which supports this scenario.

1 INTRODUCTION

IT systems pervade our daily life – at work as well as at home. Public administration or enterprise organization can hardly be managed without IT systems. We come across devices executing software in nearly every household. Increasing size (in terms of lines of code) and features of IT systems have brought us to a point, where IT systems are the most complex systems engineered by mankind.

Current research areas, like ubiquitous computing, pervasive computing, or ultra-large scale systems *[17]*, want to enable the engineering of future software systems by sharing a common trend: Complex software systems are no longer considered to have well-defined boundaries. Instead future software systems – so called ***IT ecosystems*** *[20]* – are composed of a large number of distributed, decentralized, autonomous, interacting, cooperating, organically grown, heterogeneous, and continually evolving subsystems. Adaptation, self-x-properties, and autonomous computing are envisaged in order to respond to short-term changes of the system itself, the context, or a user's expectation. Furthermore, to cover the long-term evolution of systems becoming larger, more heterogeneous, and long-lived, ***IT ecosystems*** must have the ability to continually evolve and grow, even in situations unknown during development time.

This work was funded by the NTH Focused Research School for IT Ecosystems. NTH (Niedersächsische Technische Hochschule) is a joint university consisting of Technische Universität Braunschweig, Technische Universität Clausthal, and Leibniz Universität Hannover. The NTH Focused Research School for IT Ecosystems has been established in order to deal with ***IT ecosystems***. It deals with the research questions associated with ***IT ecosystems*** in three research projects: AIM, ruleIT, and LocCom.

Research project *AIM* deals with methods and tools to guarantee the functionality of a complex ***IT ecosystem*** especially when a top-down design is not possible anymore. Within AIM, adaptive information- and collaboration architectures considering independent evolution of subsystems as well as suitable control mechanisms are examined. Therefore all levels starting at the hardware level, continuing with virtualization and modeling up to interface based formal verification are considered.

Research project *LocCom* elaborates methods, concepts, and tools for decentralized ***IT ecosystems*** enabling new emergent services and guaranteeing quality of service. Thus adaptive processes on all levels from reconfigurable hardware over protocols up to modeling and inference methods are examined. Using context information in a generalized form will be crucial for this project.

Research project *ruleIT* examines the question, whether an ***IT ecosystem*** consisting of autonomous components meets its users' requirements. Within a top-down design, ruleIT derives rules during the design steps from requirements elicitation until validation. These rules will be used for verification during development as well as for validation during runtime. To achieve this, ruleIT will combine methods from software engineering and software systems engineering, extend them and adapt them towards the development of fragments of ***IT ecosystems***. This enables, that ***IT ecosystems*** remain dependable and controllable despite the inherent autonomy of their parts.

In order to show their research results more vividly, each research project provides parts of a joint demonstrator which is a prototype of an ***IT ecosystem***. This report describes an exemplary sequence of events, which is covered by the demonstrator. This exemplary sequence is called *scenario* in the following.

The report is structured as follows: In Section 2 it provides an overview of the scenario by accompanying typical inhabitants of a smart city like New Songdo *[27]* during their travel. We will show, which benefits they gain from an ***IT ecosystem***. Moreover we relate our research questions to the scenario by introducing them and showing their impact on the scenario in Section 3. For each research question we mention, which research project deals with the question.

The report ends with a short conclusion discussing, what is needed in order to realize a demonstrator supporting the scenario.

2 SCENARIO: VISION OF A SMART AIRPORT

The scenario describes an exemplary sequence of events on a usual day at an airport like Frankfurt Airport *[1]*. We assume that an ***IT ecosystem*** is established at this airport, consisting of several IT components and subsystems. We will accompany Bob, Anna, and Chris during a travel to show the benefits, they would gain from an ***IT ecosystem***. Within the NTH Focused Research School for IT ecosystems we will develop a demonstrator which will enable the scenario presented here to show the impact of our research results.

In the scenario the protagonists Bob, Anna and Chris use small devices called ***SmartFolks***. ***SmartFolks*** can be imagined as devices with some computing power like PDAs. The ***SmartFolks*** themselves represent their owners within the ***IT ecosystem*** and act as an interface to the ***IT ecosystem***.

Step 1 (Journey to the Airport). While the first protagonist named Anna is leaving her home, her ***SmartFolk*** reminds her as she closes the door that she forgot some things. Due to sensors in the drawer of her desk the ***SmartFolk*** detects that she left her identity card there and reasons that both the passport and her travel documents are there too. In general, the sensor system is able to work with all kinds of objects which Anna has defined within her reminder list. On the way to her car she suddenly remembers that she wanted to buy some sunglasses. After a quick look at her wristwatch she decides to catch up on it at the airport and therefore adds the glasses to the ***SmartFolk***'s shopping list.

Step 2 (Parking at the Airport). The flight itinerary is available on Anna's ***SmartFolk***. Hence, the ***SmartFolk*** knows her departure terminal. As Anna is on her way to the airport the ***SmartFolk*** guides her to a parking lot that is conveniently located in due consideration of her flight details. The airport system takes care that not all ***SmartFolk*** users are transferred to the same free parking lot and that they will have free access route. Anna chooses a different parking lot than the suggested one; consequently, the system recognizes the discrepancy and asks Anna to give reasons for that. Anna gives the feedback that she chose a parking lot in the shadow as it is a very sunny day.

Step 3 (Traffic Accident). Chris is also driving to the airport while a traffic accident occurs on the streets near his current location. The accident blocks the entrance to one of the parking garages. Observation systems, e.g., ***SmartCameras***, integrated in the car and in the airport infrastructure notice the accident and send a distress signal to the Traffic Management Center (***TMC***). The information about the accident is broadcasted and spread amongst other system components. After the ***TMC*** has received and processed the message, it reacts accordingly by adjusting and redirecting traffic. Chris, located in the immediate vicinity of the accident, follows the new directions stated by his navigation system and arrives at a different parking garage.

Step 4 (Orientation). As Anna arrived at the airport, the ***SmartFolk*** leads her to a ***SmartBase*** in her vicinity. ***SmartBases*** are displayless and interfaceless sources of information spread across the airport. Compared to classical InfoKiosk or PointOfSale systems (e.g., ticket machines) ***SmartBases*** need much less and simpler components leading to lowered costs, less energy usage and far more resilience against vandalism. Thus it is feasible to deploy larger numbers of ***SmartBases*** inside and outside of the airport. The user interface for accessing the information is provided by Anna's ***SmartFolk*** which communicates wirelessly with a ***SmartBase***. Not all ***SmartBases*** are connected to a backbone network, some might even lack an electrical feed using some form of energy harvesting instead. The ***SmartBases*** hold a plethora of information queryable by Anna: Duty formalities, real estate offers, classifieds in general, flight&train schedules, etc. While Anna is accessing information relevant to her, the ***SmartFolk*** also downloads some bits of information which are not requested by her. This "parasitic" information will be automatically uploaded to other ***SmartBases*** as Anna passes them. After some time the ***SmartFolk*** will silently delete the "parasitic" information due to expiry criteria.

Step 5 (Transportation Request). At an entrance of the airport, Anna requests transportation using her *SmartFolk* and waits for an autonomous transportation vehicle (*SmartTransport*), to bring her to the designated check-in desk. However, at the same time, several large groups of travelers arrive at the train and bus station near Anna's entrance and are moving towards her position. She does not know that, at this moment, most *SmartTransports* are at a location far away from this entrance, and, by coincidence, the majority also reports a low battery power level and needs to visit a recharge station soon. Noticing the growing crowd of travelers at her location, Anna is surprised that after a short while, a sufficient number of *SmartTransports* is arriving to cope with the waiting passengers.

Step 6 (Shopping during Waiting Time). Using her *SmartFolk* Anna has written down on her shopping list that she needs sun glasses. While she is at the airport the *SmartFolk* compares the entries on her shopping list, with proposals made by shops that are near to Anna.

The sensors detect that Anna is either on the escalator or on the moving walkway. The *SmartFolk* offers two possibilities for the next steps: First, go shopping and then eat something. Second, eat something and then go shopping. Both possibilities are suggested via video and Anna can choose the option according to her preferences. The feedback of interviews like those from several *SmartFolk* users are evaluated statistically.

Bob, another *SmartFolk* user in the airport, does never react to the advertising of duty-free shops. At an interactive request he responds that being on business trips he has no time to go shopping. The *SmartFolk* developers consider this statement as a candidate for a rule. Because this is also mentioned by other people the developers integrate a new rule into the system: For traveling businessmen do not consider the way to duty-free shops.

Step 7 (Waiting Time, Goods Transport). While Anna is still waiting for the check-in, she observes the autonomous transport and delivery of goods to a nearby airport shop. Several transport vehicles have to pass a narrow opening along their way concurrently causing a small congestion. The vehicles organize and coordinate themselves, so the waiting time is spread evenly among them (see section 3.2).

Step 8 (Check-in). Now, Anna is joining the queue for the check-in desk but a tourist party blocks her way. Fortunately, she arrived early and therefore is not in hurry. However, a married couple next to her has only five minutes left to check in. The *SmartFolk* tells Anna to step aside. The tourist party steps aside as well and let the couple go to the desk.

Step 9 (Baggage Drop-off). Going away from the check-in desk Anna asks herself how her baggage is now transported over the airport. The transportation of baggage is done by an autonomous transportation service. A variety of *SmartTransports* of different sizes performs this task by self-organization. The baggage items must be carried between different locations in the airport like check-in desks, baggage security check stations, start and landing zones of airplanes, etc. Additionally, there are observation systems (e.g., *SmartCameras*, sensors, RFID readers) placed around the area, which gather and provide information (e.g. the current traffic volume) and possibly changing requirements or arising disturbances. This information is used by the *SmartTransports* (in terms of self-organization and interaction) in order to achieve a good performance of transportation.

Step 10 (Waiting Time). After checking in Anna is bored waiting for her flight. She walks around the airport hall and passes some info points which are placed everywhere on the airport. One of these info points shows on his display ideas for improving the check-in devices and provides the possibility to add own ideas. Watching some clips of other passengers Anna gets a better idea: With the help of her handbag Anna reenacts like she puts luggage on a conveyor at the check-in counter below instead of lifting it. In the past she was often annoyed with this issue. With her *SmartFolk* Anna films her action and, after this, sends the clip to the info point. Now, her clip is shown as alternative part within the check-in film. After a specific period of time the developers of the check-in devices download the passengers' ideas from the info points and have a lot of new and inspiring proposals they can realize.

Step 11 (Passport Check). Now, Anna decides to go to the gate of her flight. To reach this area she has to pass the passport check where she holds her passport beneath a small device. Briefly afterwards a green lamp flashes up, the turnstile before her is released, and Anna passes the check point. In a queue beside her Anna recognizes how another traveler has some problems and after his third illegal try an alarm sound starts and a security man comes along.

Step 12 (Waiting Time). After passing the security check Anna has to wait an hour until boarding. In order to use the waiting time meaningfully, she decides to search for more information concerning her travel destination. The *SmartFolk* recommends sights and presents photos along her travel route. These pictures are partly from public sources (e.g., *www.flickr.com*) and partly from passengers currently arriving from there. Of course, the participants do not want to share their private photos, thus intimate pictures are not sent to Anna. Nevertheless with this information Anna gets a nice overview of the sights she definitively wants to see.

Step 13 (Boarding). After some time of waiting, Anna boards the airplane. Due to the dimensions of the airport, she has to take another *SmartTransport* from the gate to her plane. As previously stated in step 3, the airport contains a *TMC* for traffic management and control inside the airport (The norms and additional traffic rules must be defined by the *TMC*, which can be considered an "Organization"). After Anna's airplane is taking off, a broken autonomous vehicle or obstacle has been detected by the *SmartCameras* installed on the bus and around the airport which blocks the first established route.

Step 14 (Departure, Travel Time, Returning). During Anna's journey the airport system is enhanced whereas the system architecture and the application itself are maintained. Amongst others, an expert system module is integrated because of the more and more fine-grained rules: No advertisements for duty-free shops are displayed to traveling salesmen except this person is inside the shop or has enough time (see Step 6). Now, being as a traveling businessmen at the airport Bob does not get any advertising of the duty-free shops.

During Anna and Bob are traveling, Chris returns from his journey. Because of a very profitable offer he bought a newly developed *SmartFolk*. Now he is curious whether the developers did a good job and whether the new device integrates itself without any problems into the *IT ecosystem* of the airport.

Step 15 (Catastrophe). A catastrophe exercise was conducted and filmed by the security cameras. The participants were interviewed afterwards to identify if the existing system acts as they expected. One criticized aspect was that participants who want to rescue victims were evacuated first and afterwards they had to go in again against the flow of refugees. After the analysis the application was enhanced according to the participants needs.

The new version of the *SmartFolk* is enhanced by an evacuation application. In case of a catastrophe only the evacuation application is available. This application provides two configurations the Evacuation and the Helper configuration. Hence, a *SmartFolk* user now has the possibility to choose two rescue relevant configurations either the Evacuation or the Helper configuration. Analogous the user chooses the Evacuation configuration in case he wants to ensure his own life, while he chooses the Helper configuration in case he decides to save the life of as many people as possible.

There is a catastrophe at the airport. A plane crashes in the waiting hall of Terminal A. A fire breaks out. All software agents located at the airport are informed; consequently the *SmartFolk* provides the evacuation application.

Chris is close to the waiting hall of Terminal A. His new ***SmartFolk*** offers Chris the two configuration possibilities provided by the evacuation application. The first opportunity is to get information regarding his evacuation and the second opportunity is to help injured persons. Chris decides to help injured people and is directed to the first patient.

Step 16 (At the train station). In the meantime, Bob isn't affected by this catastrophe. He just wants to get his connecting train as fast as possible. Certainly, this aim is shared by all travelers. The system detects who has to come first and ensures the minimal property by means of verification that nobody misses his train. While the mass around Bob starts moving the ***SmartFolk*** calms Bob down and informs him that he still has a bunch of time until his train arrives.

Step 17 (Return Journey). As Bob's train enters the station his ***SmartFolk*** recognizes the new context and shifts its environment profile from "silent" to "mobile", i.e., the vibration alarm is activated and the volume of the ring tone is increased.

3 ASSOCIATED RESEARCH QUESTIONS

Each Step of the previous section covered one, multiple or only a part of a certain research question addressed by the three research projects. The scenario showed in a demonstrative way how and where these questions interact with each other.

In this section these different research questions are described. To keep the connection to the scenario the description of each research question refers at least to the associated scenario Steps. At the beginning of the section, Table 1 gives an overview of the associations between the research questions and related scenario Steps.

		Scenario Step																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Research Questions	3.1					X		X		X				X				
	3.2					X		X		X				X				
	3.3					X		X		X				X				
	3.4					X		X		X				X				
	3.5					X		X		X				X				
	3.6			X		X		X		X				X				
	3.7		X				X				X					X		
	3.8														X			
	3.9						X					X						
	3.10															X		
	3.11								X								X	
	3.12				X	X		X										
	3.13	X					X											X
	3.14												X					
	3.15																	
	3.16																	
	3.17																	

TABLE 1. OVERVIEW OF ASSOCIATIONS BETWEEN RESEARCH QUESTIONS AND SCENARIO.

3.1 Christian Müller-Schloer, Aret Duraslan – AIM

Keywords: Coordination/Cooperation, Equilibrium, Local vs. Global Goals

As already indicated in section 1, *IT ecosystems* are composed of a large number of autonomous, decentralized, heterogeneous subsystems (agents). These systems are coupled in various ways, e.g., over the resources they use or over the goals they have. In the considered scenario, different transportation systems for people, goods, and baggages are coupled over the shared resources like roads or energy source.

If the subsystems are fully autonomous and decide on their own actions independently, then the global behavior emerges as a side-effect of the interactions of these subsystems. But we want our ***IT ecosystem*** to show the desired global system behavior, not any arbitrary system behavior. The research question then becomes how to design the interaction mechanisms concerning the coordination and cooperation between the subsystems or how to influence the local (agent) actions towards global goals to achieve the desired global behavior. In this context, another key research question is deciding the proper balance between local and global goals, since they can be to some extend in conflict. We have to evaluate the trade-offs between these bottom-up/top-down approaches in order to find an optimum between them.

In this working package, we will investigate different coordination and cooperation mechanisms, which are required to produce the intended system behavior. That satisfies both the global (system-level) goal and local (agent-level) goals at the same time. This investigation requires the analysis of different equilibrium states, which result from the interaction between the agents, and their effects on the system performance.

3.2 Jörg Müller, Christopher Mumme – AIM

Keywords: Group Decision Making, Coordination, Conflict Management

From scenario Step 5, the need for an autonomous transport system arises. At an airport, resources, such as transport network capacities, are limited: e.g., there is a maximum number of available autonomous transport vehicles. Careful planning is thus required. In the simplest case, the system is not adaptively adjusted during its runtime; rather, it is, e.g., periodically replanned (and subsequently implemented) in an offline fashion, analyzing the respective needs and possibilities. In an IT ecosystem, this mechanism is no longer sufficient, as e.g., short-term external influences cause a sudden change in transportation demand (e.g., an airplane has to make an unscheduled stop).

So the question arises, which mechanisms for decision making are helpful when handling conflicts of autonomous units. We investigate mechanisms and frameworks (e.g., InteRRaP ***[26]***, 3T ***[10]***) for coordinating decisions in dynamic environments as well as with incomplete or incorrect local models. This way, disturbances should be recognized and handled, initiating a return to equilibrium. Furthermore, we want to investigate how the behavior of agents has to be developed in connection with the world/environment, so as to support the overall system.

For instance, conflicts between transport units competing for resources may arise in an autonomous transport system. Three examples featuring different interesting aspects are shown in Figure 1:

Local conflicts, few agents: A vehicle drives against a one-way street, while another vehicle drives towards it. The scope of local conflicts is clear, so that the involved agents can find a solution.

Unknown cause, many agents: A traffic jam arises and its cause is unclear (not observable) for the agents. Because, e.g., a blockage of the road can permanently interrupt operations, the agents must find a solution for dissolving the traffic jam. The question arises whether communication-/plan-based approaches for conflict resolution are scalable or whether other, e.g., reactive procedures are more efficient.

Organized Locality: The agents move in an area, in which certain institutional norms and regulations exist. These are traffic rules in the form of, e.g., the obligation to drive on the right-hand side of the road and one-way street regulation.

In WP4 we shall start by studying the former two types of conflicts, while investigating the role of institutions in conflict resolution is an issue later in the project.

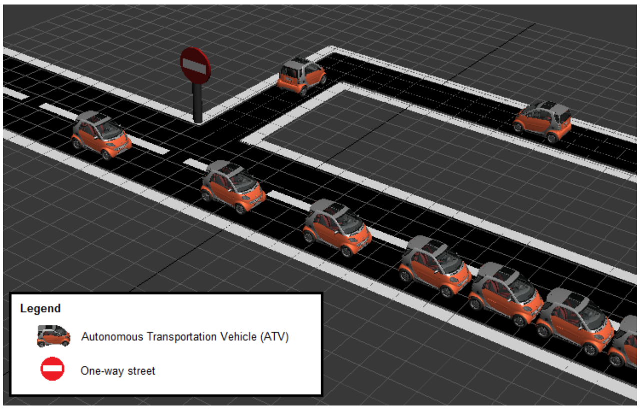

FIGURE 1. TRAFFIC JAM AND ONE-WAY STREET.

3.3 Bernardo Wagner, Christian Schulz – AIM

Keywords: Self-Organization, Interaction, Adaptivity

In the scenario Steps 5, 7 and 9, ***SmartTransports*** are used for the autonomous transportation of people, goods, and baggage. On the one hand, the execution of this task has to be as autonomously as possible (for the purpose of adaptability and flexibility), on the other hand the global goal (e.g., a high transportation performance) has to be fulfilled. Therefore, it has to be considered, how such a trade-off can be achieved. This leads to a plenty of research questions about the principles of self-organization in combination with the implicit representation of goals. According to this, the design of the system has to be somewhere in-between a central and a decentralized architecture. Considerations of the balance of these two approaches are connected with the issue of mapping the 'macroscopical' goals onto the bottom levels, where they can be realized via a (bottom-up) self-organizing process. In this context, the required information implies research on the mechanisms of measuring equilibrium and emergence. Additionally the influence of various degrees of interaction between the autonomous transport robots (exchanging information about the environment, distributed learning, development of a proper language, etc.) on the success of the self-organizing process should be examined. Further investigation concerns the problem until which degree of complexity a pure adaptation of the systems functionally is sufficient to fulfill the requirements and if an extension of the adaptation capabilities (e.g., structural self-adaptation) is necessary in some cases. Finally, the question of transferring the structures of self-organization interaction, which are successful in small scenarios directly into large-scale scenarios, is also of great interest.

3.4 Barbara Hammer, Bassam Mokbel – AIM

Keywords: Data Mining, Time-Series Clustering, Topographic Mapping, Self-Organizing Map

The coincidental development described in Step 5 of the scenario, shows a characteristic progression of system variables and monitored data (e.g., from observation systems like ***SmartCameras***, sensors, etc.), which is likely to cause an imminent imbalance in the system. In this case, considering the increasing visitor numbers near Anna's position and, at the same time, the majority of ***SmartTransports*** being at distant locations with low battery levels, a forthcoming shortage of transportation capacity around Anna's location can be expected. Our goal is to provide automatic methods, which extract such information from collected sensor data and which, on a higher level, allow for the identification of relevant system states and meta parameters from heterogeneous data provided within the ***IT ecosystem*** at any level of abstraction. In this context, data displays characteristic properties such as heterogeneity, high dimensionality, different spatial, and temporal resolution, due to the nature of the underlying process, such that uniform mining methods, which are able to deal with such data, have to be developed.

One of our research interests lies in the clustering and visualization of the available high-dimensional spatiotemporal data, by means of topographic mapping. A similarity-based representation and abstraction helps experts to explore the database intuitively and identify characteristic developments of time-series data, like the ones described exemplarily. In this regard, the question arises, which (dis-)similarity measures are useful for the different kinds of data coming from the system. However, due to the non-Euclidean nature of the proximities resulting from many of these measures, we want to investigate the application of recent extensions of common clustering methods on time-series datasets, in particular the *Relational* *[18]* variants of the *Self-Organizing Map (SOM)* and *Neural Gas (NG)*. Moreover, the usefulness of the *patch*-approach for SOM and NG *[19]* has to be considered in the context of ***IT ecosystems***, since very large input datasets are to be expected.

Time-series clustering lays the foundation for our research interests regarding classification: the assignment of the latest ongoing progression of time-series data to known characteristic developments (e.g., emergent system states) from historically recorded data. By establishing this technique during runtime, a prediction might become possible, which enables other methods to produce stabilizing reactions in order to compensate unwanted behavior of the ***IT ecosystem*** automatically. In the context of the transportation task of our fictional scenario, it would therefore become possible to automatically initiate strategies to cope with the massing of passengers at Anna's entrance (e.g. by giving ***SmartTransports*** a priority to serve clients at her location, or introducing more vehicles). Another open topic of research is the automatic training of the dimensions' relevances in the context of the named relational clustering approaches. Concerning the scenario, this so called *Relevance Learning* could determine, which system variables are crucial in characteristic situations and progressions.

In summary, the described research is relevant for various Steps in the given scenario, particularly (but not exclusively) the autonomous transportation processes in the Steps 5, 7 and 9.

3.5 Ursula Goltz, Christoph Knieke – AIM

Keywords: Metamodeling, Formal Semantics, Model-Driven Software Development, Evolution, Abstract State Machines

Our research project focuses on model-based analysis and design of ***IT ecosystems*** that are composed of a large number of subsystems. Each subsystem has to provide different levels of flexibility that can be classified as adaptability, modifiability, and evolution. By means of the transportation request Step, the goods transport Step, and the baggage claim Step in the scenario (Steps 5, 7 and 9), we will specify requirements to a modeling language for software engineering supporting the adaptation, modification, and evolution of subsystems within an ***IT ecosystem***. At first, we have to identify and clarify the particular facets of adaptation, modification, and evolution in the context of ***IT ecosystems***. Hence, we will observe use cases of the scenarios in detail dealing with these properties:

Adaptation: A system of ***SmartTransports*** has to react appropriately on exceptional circumstances such as a sudden high demand on ***SmartTransports*** at some location while some ***SmartTransports*** are currently not available. *Modification* especially concerns changes on the structure of the system: The system might be extended by additional ***SmartTransports***, or two systems of ***SmartTransports*** operating independently on different floors of the airport might be merged. The third degree of flexibility – *evolution* – focuses on fundamental changes on the system including both, modification of the structure and the behavior of the whole system: An extended generation of ***SmartTransports*** serves for both, transportation of passengers and goods, and has to be seamlessly integrated in the existing system.

Additionally, the transportation of passengers shall have a higher priority than the transportation of goods.

We will investigate how these properties can by supported adequately by model driven software development. Thus, we aim at analyzing and extending existing modeling languages for software engineering with regard to the suitability to ***IT ecosystems***. At first, we will focus on the Unified Modeling Language (UML) *[28]* as the de facto standard in model-driven software development.

The most challenging property – evolution – includes the modeling of the systems architecture and the behavior of the system. An evolving system architecture defining the structural requirements which predefine the desired degree of variability might be specified using the UML. However, we expect the UML not to be sufficient if one aims at evolving behavior models. Thus, besides this graphical approach for modeling structural and behavior information, we also regard Abstract State Machines (ASM) *[11]* which we think are more suitable to model evolutionary aspects of the systems behavior.

Finally, we will provide a metamodel *[34]* which aims at the modeling of systems with a high degree of flexibility as mentioned above. As the metamodel only covers the abstract syntax of a modeling language, we also aim at providing a formal semantics as a basis for tool support and formal verification.

3.6 Jiri Adámek, Michaela Huhn, Gianina Gănceanu – AIM

Keywords: Formal Verification, Component-Based Design, Self-Organizing Open Systems

In the scenario Step 3, we have considered the airport infrastructure to be equipped with ***SmartCameras*** that are able to identify a possible accident or an obstacle on the road which influences the normal behavior of the traffic, leading to a congestion or even more accidents (Ultimately, passengers might miss their flights or delays in baggage transport are caused, so bad feedback from clients can be expected, etc.). After the accident has taken place, the ***SmartCamera*** will send information (for example an emergency signal) to the Traffic Management Center (***TMC***), which is responsible for the control and maintenance of an efficient traffic flow in the Smart City, and to the neighbor ***SmartCameras***. The information will be routed further by the ***SmartCameras*** to the other components in the system (***SmartCameras***, PDAs, GPS navigation systems, cars equipped with in-vehicle devices which are able to interpret this information – P2P communication (in case of cars: Car2Car communication). As a result to this information the ***TMC*** will adjust the state of the road components, in order to redirect the traffic flow. For example, the traffic light in the previous intersection will be turned to red.

Based on this scenario we have identified the need to formally assure the correctness of cooperation and coordination between the components participating in the system.

Our research focuses on formal verification and validation techniques to assure functional correctness of safety-related subsystems in ***IT ecosystems***. We will investigate how basic concepts of autonomous agents like individual goals and norms imposed by a superordinated control can be integrated in formal models that facilitate automated verification of critical behavior. In the first step, we will concentrate on design verification for self-adaptive system components. Later we will extend the approach by compositional techniques that support verification of modifications at run-time. A metamodel for ***IT ecosystems*** describing fundamental concepts of autonomous agents, their collaboration and organizational structure will be developed in cooperation with AP 3, 4, and 5, to provide the basis for formalization.

Technically, *interface automata* *[15]* and *sociable interfaces* *[14]* represent the subject of our research. This formalism provides built-in notions of "compatibility" of components and of refinement which makes it well-suited for modeling self-adaptive, open systems. Interface automata or a derivate will be used to model the components behavior and also their assumptions about the environment's behavior. The interaction (coordination, cooperation, or conflicts) between the components can be checked for generic compatibility but also with regard to application-specific correctness properties. Moreover, the theory of interface automata supports the automated construction of restrictions to be added on the environment to ensure a smooth interaction between critical components. In this way, interface automata cannot only detect potential breaches of the safety objectives, but also possible safe restrictions can be suggested. Another issue is the refinement of components or services that may be offered for integration into an *IT ecosystem* at runtime. Our verification approach shall distinguish the "good" refinements (i.e. those compatible with the rest of the systems and its safety and liveness rules) from the "bad" ones (i.e. those violating the rules). This can be illustrated by adding a public certification authority to the scenario that will assess new components before they are allowed to substitute their predecessors.

3.7 Kurt Schneider, Olesia Brill – ruleIT

Keywords: Requirements Engineering, Rules, Validation

IT ecosystems comprise a wide variety of sensors, actuators, and subsystems. Subsystems evolve and change autonomously in order to meet user and customer requirements. In total, an *IT ecosystem* and its parts need to be useful to human stakeholders. Traditionally, requirements engineering aims at eliciting, handling, and validating requirements. In the realm of *IT ecosystems*, however, constraints and opportunities differ significantly from a traditional software project environment: There is little chance to interview stakeholders or carry out workshops for elicitation and validation. People, on the other hand, cannot distinguish subsystems easily. Thus, identifying and checking requirements calls for advanced approaches. In ruleIT, we seek to capture new requirements and limitations in the form of rules.

In this subproject, explicit and implicit approaches to requirements elicitation and validation are investigated. Requirements can be raised explicitly by a stakeholder. An explicit requirements technique should benefit from the *IT ecosystem* characteristics. For example, sensors and ubiquitous subsystems (e.g., mobile devices, public displays) are used for capturing ad-hoc and spontaneous requests. Such a request must then be contextualized, i.e. assigned to an appropriate context like a device under validation. In our scenario, a mobile phone with video can be used to record a problem or desired functionality in an *IT ecosystem* (scenario Step 14). Stakeholders can participate in improving their environment by enacting a new or desired use case and recording it on video (scenario Step 10). Videos can be sent easily. Subsequent analysis steps need to derive rules from contextualized stakeholder feedback.

Implicit requirements mechanisms attempt to derive and infer requirements, change requests, and rules from unexpected user behavior. Again, *IT ecosystem* infrastructure is used to observe user behavior. If a given subsystem is supposed to be validated or extended by new requirements, certain deviations from suggested or supported behavior can be identified. For example, a person who declines an assigned parking space might have an (implicit) change request for the recommender system (scenario Step 2). Like in the explicit requirements example above, appropriate analysis mechanisms need to be designed. Statistic, semi-automatic, and interactive techniques are proposed to derive and validate requirements (scenario Step 6). From a customer and stakeholder perspective, requirements validation and elicitation will merge into a novel opportunity for feedback and participation.

3.8 Andreas Rausch, Constanze Deiters – ruleIT

Keywords: Architecture Composition, Quality of Software Architecture

IT ecosystems consist of many systems and applications with their specific architectures. The architecture of a system or application defines the basic structure and provides the framework for further design and implementation of this system or application. Fundamental decisions made during defining the architecture have a high impact on a later developed system or application and, therefore, influence how well non-functional and quality requirements are met.

Software architectures are composed using multiple and different architectural building blocks. Architectural building blocks are elements like architectural principles, architectural patterns, or existing architectural parts. As aforementioned, the final architecture shall meet non-functional and quality requirements. Therefore, used architectural building blocks are chosen according how well they are suited to assure certain non-functional or quality properties. Each architectural building block has its specific structure and cannot arbitrarily be woven into the architecture. In this way, architectural building blocks affect the architecture by imposing structural or even behavioral constraints on it.

Chosen architectural building blocks are composed to achieve the architecture as a whole. The difficulty of the composition is not to plug the architectural buildings blocks technically together but to consider how to compose the properties or constraints. These last two mentioned aspects of the composition are the core of this research question.

To illustrate the arising problem regarding non-functional or quality properties consider the following example: architectural building block A assures a very good performance; architectural building block B assures only a medium performance but a very good reusability. Which conclusions can be made when composing A and B? Another example illustrates the problem regarding the imposed constraints: the architectural style *layers* defines that layers are arranged hierarchical and each layer may only use a lower layer and not vice versa. The architectural style *pipes and filters* defines that data is processed in a unit called filter and then transferred using a pipe to the next filter. Each of these styles defines another kind of constraint: the layers style specifies the direction of access between layers (control flow) and the pipes and filters style describes how data is processed (data flow). How can these constraints be combined when their styles are composed?

In the context of an ***IT ecosystem*** it is conceivable that non-functional and quality requirements of a single application are not made explicitly by application users but also by the surrounding ***IT ecosystem*** itself. Furthermore, the ***IT ecosystem*** could provide or even claim to use certain architectural building blocks. Thus, an approach addressing the described research question will support a software architect to check early whether his architecture meets the requirements of an ***IT ecosystem***. The research questions presented in this section will be addressed by the development step described in Step 14 of the scenario.

3.9 Ursula Goltz, Benjamin Mensing – ruleIT

Keywords: Modeling, Verification, Code Generation

Since an airport is a large and complex system of heterogeneous components, interacting with humans, it can be denoted as an ***IT ecosystem***. Inside the ***IT ecosystem*** of an airport exist a lot of safety-critical systems like the airport's security check (scenario Step 11) or complex and highly autonomous systems like transport robots (scenario Step 6). Especially these safety-relevant components should be provable correct. This is the central point of our research.

Since ***IT ecosystems*** make new demands on the classical software engineering process, the modeling techniques and the verification process have to be adapted consequently. Thus, we plan to modify or extend existing formalisms like Timed Automata *[2]* or Abstract State Machines *[11]* to deal with special requirements like dynamic behavior, adaptivity or evolution of the system. The aim of these extensions is to prove distinct properties, e.g., safety properties, for the ***IT ecosystem*** and accordingly avoid misconduct and harm. These properties are specified as rules the system always has to comply with. To reach this goal of provableness, the software engineer should be able to model the system before the implementation. This should be done using modeling techniques covering aspects like adaptivity or evolution, which we want to research. Those techniques should also contain rules as mentioned above. But since methods like model-checking *[13]* or static analysis *[25]* do not scale for ultra-large complex or evolutionary systems and testing is insufficient, we also want to consider approaches like runtime verification in a new context of adaptivity and evolution of the system's components. Furthermore, we want to analyze possibilities to automatically generate code starting from these newly developed models, accounting for verification and dynamic changeability at runtime.

3.10 Andreas Rausch, Sandra Lange – ruleIT

Keywords: Human Computer Interaction, Automatic System Configuration, Dynamic Adaptive Systems

IT ecosystems are used by plenty of individuals to fulfill their specific goals. A challenge coming along with an ***IT ecosystem*** is the necessity of automatically adapting itself to a current situation of a scenario. Consequently, ***IT ecosystems*** need to secure that their configurations do not only meet the requirements of the user but also the consistency with the users' needs according to the actual situation. In order to adapt itself accordingly an ***IT ecosystem*** has to derive system configuration alternatives, rules and the aforementioned user goals.

While a hierarchical system consists of subsystems, whose interactions are generally predictable and controllable, an ***IT ecosystem*** consists of individual systems, whose behavior and interaction is changing permanently. In general these changes are not planned centrally but rather result of independent processes and decisions of inner or outer effects on the ***IT ecosystem***.

According to this, the system may determine more than one configuration alternative that comply with the users' goals. The system may activate one of these configurations, but in specific situations (e.g., as described in Step 15) it has not the ability to choose the best configuration regarding to the users' goals. This makes a decision by the user indispensable. Hence the system returns those alternatives to the user. The user chooses one configuration and the binding takes place by the system.

We depict this in our SmartAirport scenario; assuming there was a catastrophe at the airport (Step 15), the system evaluates the configuration of the architecture according to the current situation. In case of a catastrophe each user has not the possibility to go shopping anymore. Hence, shopping relevant configurations are not available. The user has the possibility to choose two rescue relevant configurations either the 'Evacuation' or the 'Helper' configuration. Analogously the user decides to choose the Evacuation configuration in case he acts according his local goal and wants to save his own life, or he selects the 'Helper' configuration in case he persues the intention of the global goal and decides to save the life of as many people as possible. As the system has evaluated two possible configurations, it is not able to choose the 'right' configuration without a decision made by the user.

Either the local or the global goal might be desired by the user. But the local goal and the global goal need to be balanced. The system has not the ability to make this decision on its own. Consequently the user interaction is necessary. According to the user decision, either the evacuation or the helper configuration is selected and the system is connected accordingly. The following questions arise: How can the system adapt itself automatically and how can the Human Computer Interaction be fulfilled at runtime during the configuration evaluation?

3.11 Jürgen Dix, Heribert Vollmer, Michael Köster, Peter Lohmann – LocCom

Keywords: Modelling of Dynamical Groups, Cooperation&Coordination, Inference Methods, Abstraction Techniques, Formal Verification, Ensuring Minimal Properties, Fairness Constraints, Model Checking, Multi-Agent Systems, Strategic Abilities

Nowadays, community-based social networks become more and more important in everyday life. While in current social networks one has to explicitly use a computer or a PDA to communicate with a person and to exchange information, the idea of local communities in an ***IT ecosystem*** is more advanced: It is an integration of social communities in everyday occurrences. Instead of joining a community by explicitly subscribing to a group, in an ***IT ecosystem*** the user becomes a member of a community due to her interests and depending on her current context. Thus, the ***SmartFolk*** takes care of the details and offers the right information to the concrete situation. Such a broader view on communities leads to more dynamic and spontaneous groups. To be more precise, our focus consists in modeling strategic abilities of such competing groups. Such a view naturally induces new research questions: How to model such cooperating or competitive dynamic groups and how to ensure minimal properties (like fairness, privacy or safety)? How to reason about temporal and strategic properties of such multi-agent systems? And once we have an appropriate model, is it possible to deduce new information about the ***IT ecosystem*** based on these properties? Possibly by means of appropriate inference methods? Due to the quantity of participants in such social networks and the associated problem of state explosion we need new abstraction methods to handle model checking and formal verification. Also, due to the complexity of model checking these new methods have to run offline, i.e., each time the system changes significantly only a minimal number of properties should be checked again.

An instance of this general problem is depicted in scenario Step 8. The queue for the check-in desk is blocked by various groups all having different interests. Obviously these interests are highly competitive, as each group wants to come first. However, within a group the members have the same interests, e.g., being a married couple or joining the same tourist party. The task for the model checking system could be to verify that everybody gets to the desk and everybody gets the flight. While this illustration is just a small example one can easily imagine more complex situations on an airport or in an ***IT ecosystem***. Likewise, considering the Step 16, in which the ***SmartFolk*** recognizes the context of the user and determines which groups have to get their trains first, one can project this view onto fairness constraints regarding the bandwidth of a network with different participants or a communication system for sensors that recognizes the context of the ***SmartFolk*** user.

3.12 LarsWolf, Sebastian Schildt – LocCom

Keywords: Delay Tolerant Networking, Network Coding, Terahertz Communication

Today's communication networks available to general customers are fragmented and technologically diverse. There has been an steady increase in wireless standards including among others Bluetooth *[21]*, Wireless USB *[33]*, ZigBee (based on *[22]*), the ubiquitous WLAN *[23]*, UMTS and WiMAX *[24]*.

While total coverage for the different techniques is rising – especially for UMTS and WLAN networks – it is not realistic to reach an "always-on" status for any technology. While for example UMTS reception is more likely in outside areas one is more likely to encounter WLANs within buildings. Furthermore the ever increasing demand for bandwidth favors the development of new short range standards which are able to supply the desired bandwidth. An especially interesting area of current research deals with so called Terahertz networks. The vision here is that transceivers operating in the 60 or 300GHz bands achieve Gigabit bandwidth with distances of a few meters *[30]*.

This leads to a situation that in the future with multiple coexisting standards, there might exist some form of network everywhere but neither continuously nor technologically homogeneous. Users will move between various and heterogeneous isles of connectivity. For this kind of network setup mechanisms from Delay Tolerant Networking (DTN *[12]*) can be applied to provide a communication abstraction which does not rely on the assumption of a relatively static and always connected network like TCP does. DTN relies on the notion of a "bundle" which is a blob of information which is persistently stored by nodes and transferred to appropriate hops in an opportunistic way until it reaches its destination.

With strong immediate end-to-end semantics this also is a challenge for applications which have to be much more asynchronous by design. Imagine the transfer of large amounts of data: The sender might split the data into smaller chunks. The different chunks travel on different paths, since DTN nodes might be other mobile users moving in different directions and thus entering different networks. Furthermore, the original sender might not even be part of the network when the data finally arrives at the receiver. Using this naïve approach it is very likely that the original data might not arrive completely at the receiver since with mobile users offering data transporting service the amount of network churn is expected to be high. Therefore, some sort of redundancy is needed. The use of Network Coding techniques is proposed as it allows reconstruction of the original message even if chunks are missing. The challenge here is to decide how to encode and recombine the different chunks and under which circumstances transferring a chunk to another node. This needs a fairly accurate estimate of users' mobility and general network topology. The more information can be reliably inferred the less overhead and redundancy is necessary.

At an airport a lot of people with possible different communication abilities are entering and leaving the network within a short time, leading to high network churn. There are information flows from the airport to customers and between visitors. ***SmartBases*** deployed in the airport area equipped with short range communication capabilities provide shared storage space and secure high bandwidth islands of connectivity.

3.13 Michael Beigl, Martin Berchtold – LocCom

Keywords: Fuzzyness, Context Recognition

There are some challenges with nowadays context recognition systems. First, the recognition systems, which usually are classifiers or state recognizers, are monolithic systems only capable of recognizing a certain amount of classes or states to a certain degree of accuracy. The more classes or states being recognized the more complex a classifier needs to be. This coherency is not linear, but logarithmic. A solution here could be a 'divide and conquer' approach *[6]*. Secondly, a classification or state recognition is mostly crisp, so only the class or state is passed and no knowledge about how precise this detection was. How a quality of context classifications can be calculated was shown in *[5]*. The uncertainty of classifications was dealt with in *[4]* and how fuzziness and probability could be combined is described in *[6]*. The third issue is that a system that is determined through machine learning can only be as good as the training data has been. To obtain a perfect data set all eventualities need to be considered at design time, which is impossible. Therefore, the context recognition system needs to be able to adapt to changed conditions during runtime. How a system can change without destroying the original behavior was shown in *[7]*. Last, the recognizing system needs to separate the input data patterns in a semantically correct way. For instance, patterns from a 3-axis accelerometer sensor are highly correlated. This needs to be reflected through the separation function in *[7]*. In future reasoning methods need to be investigated, that can deal with uncertainty and deliver knowledge to adapt context recognition onto.

3.14 Wolfgang Nejdl, Kerstin Bischoff, Tereza Iofciu – LocCom

Keywords: User Generated (Meta-)Data, Tagging, Social Search, Information Propagation, (Hybrid) Recommendation Systems

Due the recent rise of Web 2.0 social networks and tagging platforms in particular, a huge amount of manually created metadata describing all kinds of resources as well as user interests is now available. Such semantically rich user generated annotations are especially valuable for searching and browsing multimedia resources, such as music, where these metadata enable retrieval on the newly available textual descriptions represented by tags. These tags represent quite a few different aspects of the resources they describe *[8]* and more research is needed on how these tags or subsets of them can be used effectively for search in (mobile) social networks. We will study which and how tags can be exploited for searching on various mobile devices in order to share multimedia content. Analyzing tags and discovering knowledge, e.g. identifying picture contents based on the tags given and/or generating additional metadata *[32]* (for music see *[9]*), is a promising way to enhance retrieval and recommendation. In order to provide personalized search and recommendation, accurate user profiles are essential. For this, we will experiment with building adequate user profiles by integrating different sources like user history, social metadata *[16, 29]*, and other derived contextual information.

Mobile social networks have not reached the critical mass, yet. However, due to the saliency of face-to-face social interaction in our everyday life, their importance also with respect to commercialization is growing fast. Wherever many people are present, you will find similar interests, competencies or goals that should be supported by smart it systems to ease question answering, information retrieval or simply socializing.

The ***IT ecosystem*** scenario centers on local communities, i.e., mobile social networks that spontaneously format the airport to exchange information or otherwise interact. Searching and propagating information between various persons can happen in multiple situations, for example, for getting informed about one's destination while waiting at the Gate for boarding. Hence, information about important or popular sights in the destination city can be recommended by searching locally and online for pictures taken by persons having been in the respective city. Besides investigating which photos to recommend based on tags (also in combination, e.g., with popularity, recency, context cues like type of trip, e.g., family or business), issues with motivation for sharing data and privacy concerns as well as timing and amount of information recommended have to be considered. For providing users with personalized suggestions, we need to create user profiles based on available information such as: user history stored on the device/web and contextual information, e.g., gathered from sensors.

3.15 Christian Siemers, Sascha Lützel – LocCom

Keywords: Green IT, PowerOptimization, ReconfigurableComputing, Adaptive Logic

At a time where climatic changes and global heating are on main focus, power efficient computing becomes a first class design constraint. But power efficiency is important not only by this reason. Most of the PDAs have to deal with strong limited battery power. The fantastic ***SmartFolk*** will face the same problem, and at the same time battery power will not significantly increase. The ***SmartFolk*** has to offer real-time analyzing of oncoming contexts and react to them. Analyzing of context will use a lot of limited battery power. Therefore it will be a challenge to balance power consumption and computation rate dynamically in obedience of oncoming contexts.

In close relationship to the energy saving problem, the problem of execution dimension will be discussed too. As two main stream execution dimensions exist – execution in time by microprocessors and execution in space by programmable logic devices – it is still an unsolved problem to find a programmable architecture capable of changing its execution dimension.

This is known as space/time mapping. The future architecture capable of this could map its runtime behavior from a more general approach with balanced execution e.g. for non-regular algorithms in the area of control systems to a specialized execution of only one task with excellent constraints concerning runtime and energy saving.

In this case the approach is in close relationship to the Universal Configurable Block-approach (UCB) discussed e.g. in *[31]*. Based on coarse-grained configurable blocks, often called FPFA, field-programmable function arrays, will be configured to form a sequential working machine, programmed by imperative or object-oriented languages like C or C++. On the other side, the microthread approach allows to isolate the behavior of one or some UCBs from the sequential execution and to execute a part of the program in space. This behavior may change frequently, and the research efforts will be focused on runtime behavior, specifically the energy consumption (or power dissipation), and the concrete definition of the UCBs (e.g. which functional units will be include to find a good balance between universal approach and efficiency).

3.16 Klaus Reinprecht, Mark Vollrath – LocCom

Keywords: User Demands, Long-Term Effects, Information Technology

The basic idea of human-centered design is to include the demands and wishes of the users in every step of the design process. This ensures that the product meets user demands and can easily be used by him. In accordance with this design approach, a questionnaire was developed with regard to possible applications of a ***SmartFolk***. The ***SmartFolk***, based on a cell phone, is an intelligent, learning companion which offers a multitude of information about available services and locations nearby including other users and enabling easy networking. Moreover, it adapts to the environment and the different aims and states of the user. With the questionnaire the ideas and wishes of the users concerning these possibilities are explored. This can then be used to develop those applications which meet the users demands. During this development the focus of our research will be on the user interface. It will be examined how to design the man-machine interaction in a way which is easily understood and can easily be used.

Finally, when a prototype of the ***SmartFolk*** is available, it will be evaluated in user tests and, if necessary, improved. This human-centered approach ensures usability and acceptance. The users can and will use this ***SmartFolk***. But what are the consequences of this usage? Will there be an improvement in the quality of life? Will this result in a new and interesting social networks? Of course, the question of these long-term effects are hard to investigate when the ***SmartFolk*** is only a prototype and not available for common use. However, in order to get an overall impression of the impact of these kinds of applications and technologies, existing devices and their effects will be examined. To this aim, a questionnaire study will be conducted comparing users of different technologies and services. The basic idea is to analyze how the availability of new information increases the quality of life and how new possibilities to contact people and communicate via electronic means changes the social networks of the people involved. Thus, an evaluation of the social and psychological effects of these technologies will be carried out.

3.17 Klaus-Peter Wiedmann, Dieter Varelmann, Marc-Oliver Reeh – AIM

Keywords: Systems theory, trust, acceptance, business model

The role of different actors and stakeholders within ***IT ecosystems*** is relevant from a marketing perspective in relation to particular objectives such as acceptance and trust. These two parameters span a continuum between autonomy of users and systems controllability. Within the framework of business administration and management theories the common focus lies on how companies, corporation, and enterprises work. The basic approach of ***IT ecosystems***, however, offers a broader approach and thus a well-elaborated extension in terms of adequacy of related problems and tasks.

Basically, modern information and communication technologies afford numerous new forms of business structures and processes, which cannot be penetrated easily by traditional theoretic approaches described above. This can be pointed out clearly (referring to the airport scenario) taking the closer collaboration between end consumers and users on different levels of the value chain, which provides, according to the advanced prosumer-approach, an exceeding service quality to consumers but, co instantaneously, asks for a higher (active) consumer involvement. In order to involve different groups of users to deliberately participate in a system described above, it is necessary to identify all trust enhancing forces, whose availability is mandatory for a successful utilization of such systems; trust in, and an individual perceived additional benefit as a result of the innovative technology / system are considered crucial preconditions for its acceptance.

Acceptability Research is one major field of interest at the Institute of Marketing and Management, in which the survey on consumer centric use acceptance relating to innovative technologies plays an important role. The airport scenario, identifying consequences for the everyday business activities, perfectly fits the Institute's research agenda. The main purpose of this study, by means of integrating the ***IT ecosystems*** approach in to the acceptability concept and taking quantitative as well as qualitative marketing methods, is the identification of value parameters in respect of applicability, utilization, and acceptance. Finally, traditional business models needs to be newly conceptualized to changing market necessities by means of inter-disciplinary approaches and evaluations.

4 CONCLUSIONS

In the previous sections we introduced a demonstration scenario to illustrate the research questions behind ***IT ecosystems***. These research questions will be addressed in the three research projects (AIM, LocCom, and ruleIT) of the NTH Focused Research School for IT ecosystems.

The scenario description presented here is not only "paperware". It will be further refined towards a complete scenario specification defining software and hardware components and their interfaces.

Software and hardware components will be derived from the scenario description. These are the components which need to be provided by the research projects. Since these components will interact with each other in the demonstrator, all research projects need to agree on the interfaces between these components.

This interface specification will describe syntactical aspects describing the structure of the interface as well as semantical aspects. Thus, it defines the behavior of components appearing in the demonstrator. It is essential to enable the research projects to develop these components loosely coupled while keeping the integration risk low.

The integration of the ***IT ecosystems*** from these different components enables us to show the impact of our overall research within a visionary demonstrator, which is an ***IT ecosystem***. Anyhow, it is still a long road to go until we achieve this demonstrator. The refined scenario specification will be elaborated in the following months and an infrastructure enabling the integration of the scenario parts will be elaborated.

GLOSSARY

IT ecosystem An ***IT ecosystem*** is a system composed of a large number of distributed, decentralized, autonomous, interacting, cooperating, organically grown, heterogeneous, and continually evolving subsystems.

SmartBase ***SmartBases*** are autonomous information storage systems, equipped with wireless transmission capabilities.

SmartCamera A ***SmartCamera*** or an intelligent camera is an embedded vision system that is capable of extracting application-specific information from the captured images, along with generating event descriptions or making decisions that are used in an intelligent and automated system. (Definition taken from *[3]*).

SmartFolk ***SmartFolks*** can be imagined as small devices with some computing power like PDAs. In the scenario (see section 2) ***SmartFolks*** embody their owners Anna, Bob and Chris within the ***IT ecosystem*** and act as an interface to the ***IT ecosystem***.

SmartTransport The term ***SmartTransport*** stands for a driverless, intelligent, autonomous transportation vehicle within the airport. People, baggage, as well as goods are transferred between different locations using ***SmartTransports***. Due to the variety of transportation tasks, they are not necessarily assumed to be uniform regarding their technical abilities and equipment. ***SmartTransports*** are using and sharing infrastructures of the airport like communication and transportation networks, acting by their own initiative. Hence, they can be seen as intelligent agents within the ***IT ecosystem***.

TMC The Traffic Management Center (***TMC***) is an intelligent entity (an organization in the context of ***IT ecosystems***) responsible for the management and control of traffic within the airport. Its aim is to keep the system in balance and therefore to ensure an efficient and uninhibited flow of traffic in the transportation network. The ***TMC*** defines the organization structure by specifying the roles that can be enacted by agents, the relations between the roles, the norms which refer to its members, and the location-based norms which imply a specific behavior of the agents that are located in the defined area.

REFERENCES

[1] Fraport AG. Webpage of Frankfurt airport. *http://www.frankfurtairport.com*, 2009.

[2] R. Alur and D. L. Dill. A theory of timed automata. Theoretical Computer Science, 126:183–235, 1994.

[3] Ahmed Nabil Belbachir, editor. Smart Cameras. Springer, June 2009.

[4] M. Berchtold and Michael B. Increased Robustness in Context Detection and Reasoning using Uncertainty Measures - Concept and Application. In Proceedings of the European Conference on Ambient Intelligence (AmI'09). Springer, 2009.

[5] M. Berchtold, C. Decker, T. Riedel, T. Zimmer, and M. Beigl. Using a Context Quality Measure for Improving Smart Appliances. In Proceedings of the 7th International Workshop on Smart Appliances and Wearable Computing. IEEE, 2007.

[6] M. Berchtold, T. Riedel, M. Beigl, and C. Decker. AwarePen - Classfication Probability and Fuzziness in a Context Aware application. In Proceedings of the 5th International Conference on Ubiquitous Intelligence and Computing, pages 647–661. Springer, 2008.

[7] M. Berchtold, T. Riedel, K. van Laerhoven, and C. Decker. Gath-Geva Specification and Genetic Generalization of Takagi-Sugeno-Kang Fuzzy Models. In Proceedings of the IEEE International Conference on Systems. IEEE, October 2008.

[8] K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu. Can all tags be used for search? In Proceedings CIKM 2008, pages 192–202. ACM, 2008.

[9] K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu. How do you feel about "dancing queen"?: deriving mood & theme annotations from user tags. In Proceedings JCDL 2009, pages 285–294. ACM, 2009.

[10] R. P. Bonasso, R. J. Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack. Experiences with an architecture for intelligent, reactive agents, 1997.

[11] E. Börger and R. F. Stärk. Abstract State Machines: A Method for High-Level System Design and Analysis. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[12] S. Burleigh, A. Hooke, L. Torgerson, V. Cerf, R. Durst, K. Scott, K. Fall, and H. Weiss. RFC 4838 – Delay-Tolerant Networking Architecture. RFC, 2007.

[13] E. M. Clarke, O. Grumberg, and D. A. Peled. Model Checking. The MIT Press, January 1999.

[14] L. de Alfaro, L. Dias da Silva, M. Faella, A. Legay, P. Roy, and M. Sorea. Sociable interfaces. In Proceedings of 5th International Workshop on Frontiers of Combining Systems, volume 3717 of Lecture Notes in Computer Science, pages 81–105. Springer, 2005.

[15] L. de Alfaro and T.Henzinger. Interface automata. In Proceedings of the Ninth Annual Symposium on Foundations of Software Engineering (FSE), pages 109–120. ACM Press, January 2001.

[16] J. Diederich and T. Iofciu. Finding communities of practice from user profiles based on folksonomies. In Proceedings of the 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice, 2006.

[17] P. Feiler, R. P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, L. Northrop, D. Schmidt, K. Sullivan, and K. Wallnau. Ultra-large-scale systems: The software challenge of the future. Technical report, The Carnegie Mellon Software Engineering Institute, 2006.

[18] B. Hammer and A. Hasenfuss. Relational neural gas. In J. Hertzberg, M. Beetz, and R. Englert, editors, KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI, KI 2007, volume 4667 of Lecture Notes in Artificial Intelligence, pages 190–204, Berlin, 2007. Springer.

[19] A. Hasenfuss, B. Hammer, and F. Rossi. Patch relational neural gas — clustering of huge dissimilarity datasets. In ANNPR '08: Proceedings of the 3rd IAPR workshop on Artificial Neural Networks in Pattern Recognition, pages 1–12, Berlin, Heidelberg, 2008. Springer-Verlag.

[20] S. Herold, H. Klus, D. Niebuhr, and A. Rausch. Engineering of it ecosystems: design of ultra-large-scale software-intensive systems. In ULSSIS '08: Proceedings of the 2nd international workshop on Ultra-largescale software-intensive systems, pages 49–52, New York, NY, USA, 2008. ACM.

[21] IEEE. IEEE 802.15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs). IEEE Standard for Information technology pp. 1-600, 2005.

[22] IEEE. IEEE 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-RateWireless Personal Area Networks (WPANs). IEEE Standard for Information technology, 2006.

[23] IEEE. IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE Standard for Information technology, 2007.

[24] IEEE. IEEE 802.16: Air Interface for Fixed Broadband Wireless Access Systems. IEEE Standard for Local and metropolitan area networks pp. 1-895, 2009.

[25] T. Lev-Ami, T. Reps, M. Sagiv, and R. Wilhelm. Putting static analysis to work for verification: A case study. SIGSOFT Software Engineering Notes, 25(5):26–38, 2000.

[26] J. P. Müller. The Design of Intelligent Agents: A Layered Approach. Springer- Verlag New York, Inc., Secaucus, NJ, USA, 1996.

[27] Pamela L. O'Connell. Korea's high-tech utopia, where everything is observed. The New York Times, 2005.

[28] OMG. UML, Version 2.2. OMG Specification Superstructure and Infrastructure, *http://www.omg.org*, 2009.

[29] M. Pannevis and M. Marx. Using web-sources for location based systems on mobile phones. In Proceedings of the SIGIR 2008 Workshop on Mobile Information Retrieval, pages 52–53, 2008.

[30] R. Piesiewicz, M.N. Islam, M. Koch, and T. Kurner. Towards shortrangeterahertz communication systems: Basic considerations. In Applied Electromagnetics and Communications, 2005. ICECom 2005. 18th International Conference on, pages 1–5, 2005.

[31] C. Siemers and V. Winterstein. The Universal Configurable Block/Machine – An Approach for a Configurable SoC-Architecture. The Journal of Supercomputing, 26:309–331, 2003.

[32] B. Sigurbjörnsson and R. v. Zwol. Flickr tag recommendation based on collective knowledge. In Proceedings WWW 2008, pages 327–336, 2008.

[33] USBIF. Wireless Universal Serial Bus Specification. USBIF pp. 1-303, 2005.

[34] H. Vangheluwe, J. de Lara, and P. J. Mosterman. An Introduction to Multi-Paradigm Modelling and Simulation. In Fernando Barros and Norbert Giambiasi, editors, Proceedings of the AIS'2002 Conference (AI, Simulation and Planning in High Autonomy Systems), pages 9–20, 2002.