IT Ecosystems: A new Paradigm for Engineering Complex Adaptive Software Systems

Andreas Rausch, Jörg P. Müller,
Dirk Niebuhr, Sebastian Herold
Technische Universität Clausthal
Email: firstname.lastname@tu-clausthal.de

Ursula Goltz
Technische Universität Braunschweig
Email: goltz@ips.cs.tu-bs.de

***Abstract*—Today's software-intensive systems are among the most complex artifacts created by men. This is due to ever increasing requirements and functionality of the software on the one hand, and to rising structural complexity with respect to size, interconnectedness, and distribution on the other hand. Engineering and controlling these systems pushes existing software engineering approaches to (and beyond) their limits [1].**

This paper describes the concept of IT ecosystems as a new approach for addressing this challenge from the perspective of software engineering. The concept and approaches described were developed in a large interdisciplinary research project (www.it-oekosysteme.org); we present first results including a validation scenario of a smart airport, which has been devised and implemented in the project, aiming at a comprehensive approach to IT ecosystems engineering.

I. Introduction

Software now pervades all areas of work and society. Public administration, management, organization and production companies as well as day-to-day personal life are no longer conceivable without the use of software. Software-controlled devices can be found in every household.

The continuous increase in size and functionality of software-intensive systems [1] have now made them among the most complex man-made systems [2]. The reasons for the steady increase in their complexity are twofold: On the one hand, the set of requirements imposed on software-intensive systems becomes larger and larger; the *extrinsic complexity* increases. This includes, for example, features, depth of functionality, adaptability, and variability. On the other hand, the structures of software-intensive systems, e.g., in terms of size, scope, distribution and networking of the system, are themselves becoming more complex; this leads to an increase in the *intrinsic complexity* of the system.

The expectations in software-intensive systems have been growing and continue to do so with their steadily increasing penetration into people's private, social, and professional lives. Buyers and users of these systems expect:

- A significantly higher flexibility, adaptability, intuitive usability and timely response to changes in both the software system itself as well as in the processes for the expected life cycle and demands.
- A high degree of reliability (Dependability [3]) of the software system and the surrounding development, operation, and administration processes.

In the long run, the continuously growing complexity of software-intensive systems, and the rising user expectations have led to a situation where the classical methods and techniques of computer science reach their limits. As an analogy, let us consider the field of classical engineering: There, a single (even large) building can still be planned, explained and implemented centrally; however, the planning, establishment and development of a city need to be performed using very different methods and models. Similarly, the mechanisms required in computer science to develop and control software-intensive systems is also facing a paradigm shift.

To react to this challenge, in this paper we put forward the proposal to interpret software-intensive systems as part of a larger IT ecosystem, thus leading a step in the direction of such necessary paradigm shift. In the NTH School for IT Ecosystems (www.it-oekosysteme.org), we are involved in a comprehensive research program on concepts, architectures, platforms and tools to enable and support this paradigm shift.

The main contributions of the paper are the definition of a conceptual model of an IT ecosystem, and the specification of two scenarios: a general system scenario addressing some generic important properties of IT ecosystems, and a specific validation scenario, a smart airport. The paper is structured as follows: Basic characteristics of IT ecosystems are defined in Section II. Section III discusses the conceptual core components of the IT ecosystems paradigm in more detail. Two validation scenarios are defined and discussed in Section IV. The paper ends with a conclusion and outlook to future research in Section V.

II. Characteristics of IT ecosystems

IT ecosystems are a class of systems that obey to certain characteristics and fulfil certain requirements. In analogy to the concept of an ecosystem in biology, IT ecosystems achieve reliability by means of some higher-level regulation system, through which they maintain equilibrium between the forces applied by the participating individuals. It is the balance between *controllability* of the whole ecosystem, and the *autonomy* of the system participants is the key characteristic of an IT ecosystem. When this balance is disturbed, the IT ecosystem breaks and it is no longer manageable. For an IT ecosystem to remain active and continuously evolve, we must understand this balance and the mechanisms necessary to achieve and maintain it.

A number of key research questions need to be addressed in this context: What kind of systems can be regarded as useful IT ecosystems? How can you recognize an IT ecosystem, i.e., how can you decide for a given system whether it is (or: should be conceived as) an IT ecosystem? Obtaining systematic and scientifically-based answers to these questions is ultimately a goal of NTH School for IT Ecosystems.

IT ecosystems are complex adaptive *Systems of Autonomous Systems* — i.e., complex system compounds consisting of interacting autonomous individual systems, which are adaptive as a whole, based on local adaptiveness (see outer ring in Figure 1). This means that not every large system can be considered as an IT ecosystem: The complexity of the interaction between the IT ecosystem and the resulting adaptivity is central to the autonomy of individual systems. It must also consider different life cycles of the individual systems.

978-1-4673-1703-0/12/$31.00 ©2013 IEEE

Herein lies an important difference from the traditional understanding of hierarchical systems: A hierarchical system consists of subsystems, whose interactions tend to be globally predictable, controllable, and designable. An IT ecosystem is composed of individual autonomous systems whose behavior and interactions change over time. These changes are usually not centrally planned, but arise from independent processes and decisions within and outside the IT ecosystem.

In addition, IT ecosystems are mixed human-machine artifacts: humans in the IT ecosystem (see the inner ring in Figure 1) interacts with the individual systems in the IT ecosystem; this way, humans become an integral, active part of the IT ecosystem system – their requirements, goals, and behavior must be considered by modeling them as autonomous system components. Humans act as users, administrators, operators within the IT ecosystem. The very complex and multifaceted interaction and relationship between people and individual systems of an IT ecosystem is a key aspect. Only by including this aspect, a holistic approach can be established. The requirements, needs and expectations of humans in the individual systems of an IT ecosystem are subject to special dynamics and interaction. Thus, the individual systems need to be able to change continuously to meet the changing demands and adapt to changing behavior of humans. On the other side, changing expectations of humans will create new demands and needs.

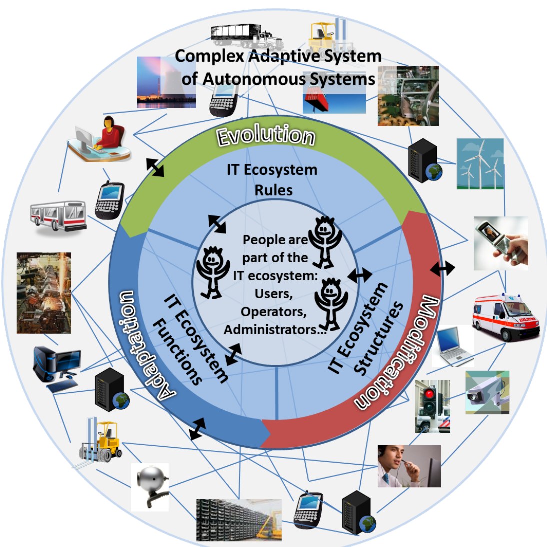

Fig. 1. Structure of an IT ecosystem – autonomy and controllability

III. IT ECOSYSTEM AS A PARADIGM OF AUTONOMY AND CONTROLLABILITY

As previously discussed, an IT ecosystem is made up of autonomous individual systems designed to interact with each other. Except for these interactions, the individual systems are considered as closed systems that can be created with the classical methods of software development and validation. However, in doing so, adaptivity, evolution and autonomy must already be considered. The individual systems themselves may consist of subsystems or components, or are used as sensors, actuators, or the interface to a physical environment.

The system compound as a whole can no longer be described and controlled by using classical methods. In addition to the complexity caused by the size of the system compound and its adaptability due to the autonomy of individual systems and their different life cycles (see Section II), humans are considered as a part of the IT ecosystem, too. The resulting IT ecosystem can be described and understood only by taking a holistic view. This is a necessary condition for the controllability of the overall system. However, this holistic approach leads to a very complex system with a high degree of autonomy, which in turn makes it difficult to control.

This leads us to a dilemma: In order to control the system, we need to look at it holistically; doing so increases the degree of autonomy, which in turn reduces controllability. To solve this dilemma, we must turn to the notions of autonomy and controllability in the IT ecosystem.

We distinguish three levels of autonomy in an IT ecosystem (see the middle ring in Figure 1). It should be noted that the higher the degree of autonomy, the stronger the human is involved in this autonomy:

1) *Adaptation* is an ability of an IT ecosystem: They provide mechanisms to ensure their short-term autonomy. By adaptation, we mean the property of the compound system and its autonomous individual systems to reconfigure and reorganize themselves in order to fulfil context-sensitive tasks in the system. Adaptation is therefore the short-term and often preplanned capability of individual components and their interaction to adapt. Here, primarily the functionality is concerned, as shown in Figure 1 in the middle ring; adaptation is often achieved by modifying component configurations – parameters are set which alters the functional behavior of system components.
   Individual systems may consist of (semi-)autonomous components, which we refer to as agents [4]. Adaptivity is necessary because the tasks to be performed vary greatly, and may be in conflict with other tasks, or because availability or functionality of the agents changes. These tasks are preformed in a (semi-)autonomous manner through coordination or cooperation between the agents [5]. During task performance, plans are made and plan status and viability must be constantly checked. If deviations are detected, it may be necessary to reschedule [6].
2) By *Modification* we understand the ability of an IT ecosystem to provide short- and medium-structural adaptability which is grounded in the autonomy of the individual systems. IT ecosystems are open and dynamic systems: new components and individual systems may enter into the system, with possibly unknown interface structure and behavior. Already known components and individual systems may change their behavior or leave the It ecosystem. Thus, modification means adapting functionality *and* structure can be adapted to new requirements and constraints similar to the way proposed in autonomic computing [7] or organic computing [8], respectively. Subsystems are dissolved and new components are added. To conclude, modification is extended adaptation, since it includes both functional and structural changes, as shown in Figure 1 in the middle ring. Also humans can enter or leave the IT ecosystem, as may do other physical objects carried by humans, such as hardware components.
3) Evolution is the ability of an IT ecosystem to develop under changing external conditions in the medium- to long-term, and to sustainably reveal autonomous behavior. It includes the fundamental long-term development of the IT ecosystem in all its aspects, in particular through change and adaptation of monitoring, configuration, and control mechanisms, including structural and functional aspects. Evolution in terms of IT ecosystems, thus includes adaptation and modification. Evolution extends these concepts by the capability of changing the rules in the

IT ecosystem itself (see Figure 1, middle ring). Therefore, implementing evolution as manual, computer-supported, or (partially) automated further development of the IT ecosystem poses the biggest challenge with respect to long-term control and controllability. Evolution will be triggered by sustainable changes in environmental conditions or by fundamental changes in the expectations of users and operators of the IT ecosystem. It can be driven by human operators and users, but it also needs to be partly or fully automated in some cases. Evolution can mean either that the management, control and regulatory mechanisms are altered, or that individual components or entire systems are replaced or modified.

These three levels of autonomy are required from all the participants in the IT ecosystem. However, at the same time care needs to be taken that the IT ecosystem as a whole remains under control and thus ensures its superordinated goal and function. For this to be achieved, the participating autonomous systems and components accept a set of general rules, conceivable e.g., as traffic rules in the context of a smart city, to ensure the proper functioning of the entire system.

A key aspect in the explanation of the above effects and interrelationships is the question of the existence of a balanced equilibrium between autonomy and controllability of the IT ecosystem as shown in Figure 2. The IT ecosystems approach is based on:

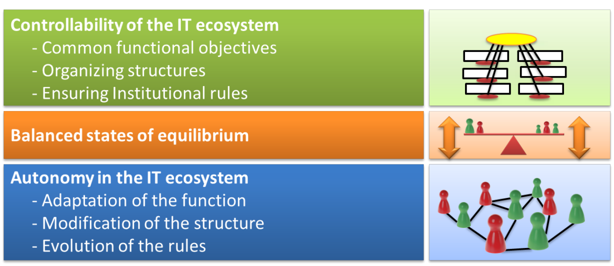

Fig. 2. Balanced equilibrium between autonomy and controllability

- the existence of equilibrium states, i.e., states that ensure a smooth functioning of the overall system (e.g., the flow of traffic in a city)
- forces that aim at disturbing the balance (such as a rear-end collision on a major road junction), and
- centralized or decentralized mechanisms to maintain or re-establish the balance (such as a central traffic control system or communications and distributed intelligence of vehicular navigation systems).

If equilibrium states can be established permanently, we have achieved the goal of providing desirable autonomy, while at the same time ensuring controllability.

To ensure controllability, IT ecosystems must feature a set of concepts outlined in Figure 2:

1) Communities of autonomous systems and individual players should form themselves dynamically. An essential feature of these communities are common and jointly accepted functional objectives. Individual systems and components can be simultaneously be members of several communities. These communities may change over time, dissolve, and new ones can be created. This is part of the adaptation of the functionality in the IT ecosystem (cf. Figure 1).
2) Structures required for organizing and implementing the functional objectives of the communuty form dynamically. These structures define roles, responsibilities, communication channels, and interaction mechanisms in the communities. Like the communities themselves, organization structures can also change, thus leading to a modification of the structures in the IT ecosystem (see Figure 1).
3) Commonly accepted rules govern the behaviour and interactions of communities and their organizational structures. Control within IT ecosystems (in a sense of ensuring adherence to these rules) can be realized by different means. An important approach in this context are electronic institutions [9]. The rules in the IT ecosystem can be changed through the concept of evolution. As shown in Figure 3, institutions should ensure management, control and regulation mechanisms – the basic rules within the organizational structure of communities, in the IT ecosystem. These mechanisms can be explicit, e. g., centralized or federated via dedicated components, or implicit, for example, realized by market mechanisms, local incentive and preference structures of individual systems or components to achieve a specific behavior of the system.

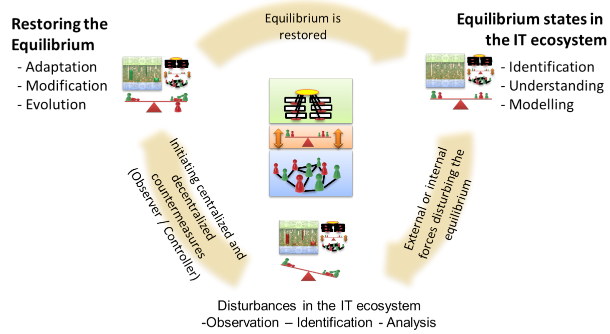

Fig. 3. Ensuring equilibrium states in IT ecosystems

The concept of equilibrium in IT ecosystem enables us to provide mechanisms for control, monitoring, and regulation, and to ensure rule compliance via electronic institutions. In case these rules are violated, the adaptation, modification, and evolution mechanisms provided by the IT ecosystem can re-establish the balance. Based on these mechanisms, equilibrium concepts are defined and approaches to detection, prevention and treatment of disorders in the IT ecosystem are described and implemented.

IV. IT ECOSYSTEMS SCENARIOS

The above properties allow an assessment of applicability and usefulness of the IT ecosystem metaphor for different system scenarios. However, it is clear that these criteria are not entirely clear-cut. There is a gray area here - we consider, for example, a large automated high bay warehouse, that include the autonomous vehicles transporting goods and merchandise orders and deliveries. Is such a system an IT ecosystem — or not?

Answering the question forces a detailed analysis of system scenarios. As a result, we in the following, we illustrate and study the notion of IT ecosystems by means of two scenarios: a generic application system / infrastructure scenario (Section IV-A, and a specific instance of the generic scenario describing a smart airport (Section IV-B).

A. System scenario: Application System + System Infrastructure

A compound system includes an application system that uses functions of an underlying infrastructure system. Application system and infrastructure system are developed and/or operated largely independent from each other, by different organizations. Changes in the infrastructure system lead to disruptions, faults, and subsequent need for action in the application system. Conversely, changes in the application system will place new demands on and require adjustments to the infrastructure.

Consider, for example, a PC-application environment based on the MS Windows operating system in a large company. Here, it is required to provide a number of individual servers such as mail server, calendar management, web server, database server, or workflow services, in order to support corporate functions and processes. This system is thus a system compound. Humans are part of the system and interact (as customers or employees) with the individual systems. Furthermore, the system is capable of *adaption* – it performs load balancing and coordinates user requests in order to deal with hot spots. The system will undergo *modification*: Employees can join or leave the system e.g. with mobile devices. The system must integrate new types of services to provide the enterprise with new funcationality.

Finally, also *evolution* takes place in this system, e.g., via automatic mechanisms for updating individual systems. The evolution and emergence of new individual systems can produce requirements for adapting, mofiying, or evolving other component systems and the infrastructure. For example, the integration of a high-definition video conferencing systems require the support of real-time transport protocols by the operating system as well as an upgrade of the corporate network to Gigabit LAN.

Required guarantees can be provided in this system e.g., via access rights (the access of certain users is restricted to parts of the system) or via service level contracts and corresponding service level enforcement mechanisms to assert users certain functions in a certain quality. In a company such as Deutsche Brse, mechanisms are established, e.g. to recognize and regulate irregular behavior such as panic selling. Since this system scenario has all the aforementioned characteristics, such systems are IT ecosystems.

B. Validation Scenario: Smart Airport

The second scenario which we propose to validate approaches for IT ecosystems, is much more specific. It 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. We have developed a demonstrator which will enable the scenario presented here to show the impact of our research results. Figure 4 illustrates the systems being parts of the overall IT ecosystem application scenario.

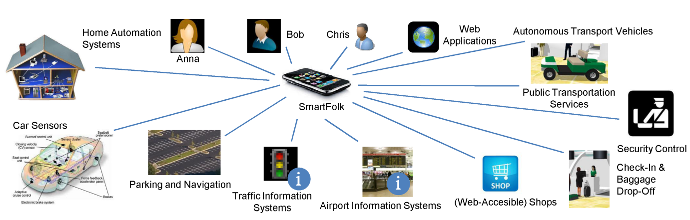

Fig. 4. Overview of the systems being parts of the IT ecosystem scenario

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.

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 her passport and travel documents are there too. The sensor system is able to work with all kinds of objects Anna has defined in her reminder list. On the way to her car she 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 adds the glasses to the SmartFolk's shopping list.
2) (Parking at the Airport). The flight itinerary is available on Anna's SmartFolk. As she is on her way to the airport the SmartFolk guides her to a parking lot conveniently located to her departure terminal. 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; 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.
3) (Traffic Accident). Chris is also driving to the airport while a traffic accident occurs near his current location, blocking 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 is broadcasted and spread amongst other system components. After the TMC has received and processed the message, it reacts by adjusting and redirecting traffic. Chris, located near the accident, follows the new directions stated by his navigation system and arrives at a different parking garage.
4) (Orientation). Upon arrival at the airport, the SmartFolk leads Anna to a nearby SmartBase. SmartBases are displayless and interfaceless sources of information spread across the airport. Compared to classical InfoKiosk or PointOfSale systems, SmartBases need much less and simpler components leading to lower costs, less energy usage and more resilience against vandalism. 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 use some form of energy harvesting instead. The SmartBases hold a plethora of information: Duty formalities, real estate offers, classifieds, flight&train schedules, etc. While Anna is accessing information relevant to her, the SmartFolk also downloads additional bits of information. 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 based on expiry criteria.
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,

[1] http://www.frankfurtairport.com

978-1-4673-1703-0/12/$31.00 ©2013 IEEE

and, by coincidence, the majority also reports a low battery power level and need to visit a recharge station. 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.

6) (Shopping during Waiting Time). Anna has noted on her shopping list that she needs sunglasses. 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: Either, go shopping and then eat something, or the other way around. 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, never reacts 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. Because this is also mentioned by other people the SmartFolk developers integrate a new rule into the system: For traveling businessmen do not consider the way to duty-free shops.
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.
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.
9) (Baggage Drop-off). At the check-in desk Anna asks herself how her baggage will be transported over the airport. This is done by an autonomous transportation service. SmartTransports of different sizes perform 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), changing requirements or arising disturbances. This information is used by the SmartTransports (in terms of self-organization and interaction) to optimize transportation.
10) (Waiting Time). After checking in Anna is bored waiting for her flight. She walks around the airport hall and passes some info points placed on the airport. One of them displays ideas for improving the check-in devices and provides the possibility to add own ideas. Watching some clips by other passengers Anna gets a better idea: With the help of her handbag Anna reenacts that 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. After a specific period of time the developers of the check-in devices download the passengers' ideas from the info points and thus gain proposals for improvement.
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. 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.
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. Pictures are partly from public sources (e.g., www.flickr.com) and partly from passengers currently arriving from there. As participants do not want to share their private photos, intimate pictures are not sent to Anna. With this information Anna gets a good overview of the sights she definitively wants to see.
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.
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 update to the rule base is installed: 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).
    While Anna and Bob are traveling, Chris returns from his journey, where he bought a newly developed SmartFolk. Now he is curious whether the developers did a good job and whether the new device smoothly integrates with the airport IT ecosystem.
15) (Catastrophe). A catastrophe exercise was conducted and filmed by the security cameras. The participants were interviewed afterwards whether 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. 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. A SmartFolk user now has the possibility to choose two rescue relevant configurations: she can select the Evacuation configuration in case he wants to ensure his own life, and choose the Helper configuration if she 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; the SmartFolk provides the evacuation application.
    Chris is close to the waiting hall of Terminal A. His new SmartFolk offershim 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.

978-1-4673-1703-0/12/$31.00 ©2013 IEEE

16) (At the train station). In the meantime, Bob wants to get his connecting train as fast as possible. The system detects who has to come first and ensures the minimal property by means of verification that nobody misses his train. While the crowd around Bob starts moving the SmartFolk calms Bob down and informs him that he still has enough time until his train arrives.
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.

The presented scenario can be seen as a IT ecosystem. *Adaption* in the abovementioned sense happens for example by the TMC directing traffic, or when Anna's SmartFolk connects after arrival to the airport's IT system. The available set of components running on a SmartFolk ist *modified* in the case of accidents (only the evacuation application is available). Furthermore, *evolution* happens when the basic rule for non-shopping businessmen is added.

Note, that the proposed approach of IT ecosystems uses this airport scenario as example only and does not compete with approaches dealing with airports or single aspects of these complex systems [10]–[16]. On the contrary, because this domain has been investigated very well, those situations, in which the levels of autonomy and contollability are required, are well understood but not supported by a systematic (and domain-independent) engineering approach.

V. Conclusion and outlook

In this paper, we have defined a new approach towards complex, software-intensive systems: IT ecosystems. Our approach combines the systems-of-systems view as defined e.g. by [17] with research on ultra-large scale systems [1], but extends these approaches by the use of the multiagent systems metaphor [6] in order to express autonomy and decentralized control. Also, by acknowledging the role of the human as part of the IT ecosystem, our approach opens up new research venues linking control theory and software engineering with human-machine interaction and psychology. Last but not least, our model borrows notions of balance and equilibria from biologically inspired ecosystems research. The main contributions of the paper are a conceptual model of an IT ecosystem, and the specification of two scenarios: a general system scenario addressing some generic important properties of IT ecosystems, and a specific validation scenario, a smart airport.

A current limitation of our work lies in the somewhat restricted suitability of the currently considered airport scenario as regards its IT ecosystem characteristics. In the terminology of Maier [17], this scenario can be best classified as belonging to the simplest class of complex systems of systems, so called *directed systems*, in which an overarching system purpose exists and the complex system is built and controlled to this purpose, with very limited long-term evolution. So far, we have not yet investigated more open, complex, and evolutionary system types, such as *collaborative* or *virtual systems*, and corresponding scenarios, which remain areas of future work. Still, as the scenario description shows, even the smart airport contains considerable complexity worth investigating, mainly introduced by including humans in different roles as part of this IT ecosystem. Future work will apply the concepts developed in this paper to other domains including urban traffic management [18] and social network engineering.

VI. Acknowledgments

This publication was supported by NTH Focused Research School for IT Ecosystems. Many of the thoughts included in this article origin from discussions with Professors and Scientific staff in the NTH Focused Research School for IT Ecosystems. Our special thanks go to Ingrid Schmees and Sandra Lange for their assistance in editorial revision of the article.

References

[1] L. Northrop, P. Feiler, R. P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, K. Sullivan, and K. Wallnau, "Ultra-Large-Scale Systems - The Software Challenge of the Future," Software Engineering Institute, Carnegie Mellon, Tech. Rep., June 2006. [Online]. Available: http://www.sei.cmu.edu/uls/downloads.html

[2] A. W. Brown and J. A. McDermid, "The art and science of software architecture," in *ECSA*, ser. Lecture Notes in Computer Science, F. Oquendo, Ed., vol. 4758. Springer, 2007, pp. 237–256.

[3] J. C. Laprie, A. Avizienis, and H. Kopetz, Eds., *Dependability: Basic Concepts and Terminology*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1992.

[4] J. P. Müller, *The Design of Intelligent Agents*, ser. Lecture Notes in Artificial Intelligence. Springer-Verlag, 1996, vol. 1177. [Online]. Available: http://www.springer.com/computer/ai/book/978-3-540-62003-7

[5] J. Vázquez-Salceda, H. Aldewereld, and F. Dignum, "Norms in multiagent systems: From theory to practice," *Int. Journal of Computer Systems & Engineering*, vol. 20, no. 4, pp. 225–236, 2005.

[6] M. Huhn, J. Müller, J. Görmer, G. Homoceanu, N.-T. Le, L. Märtin, C. Mumme, C. Schulz, N. Pinkwart, and C. Müller-Schloer, "Autonomous agents in organized localities regulated by institutions," in *Proc.of IEEE DEST 2011: 5th IEEE International Conference on Digital Ecosystems and Technologies*. IEEE Press, 2011, pp. 54–61.

[7] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003. [Online]. Available: http://dx.doi.org/10.1109/MC.2003.1160055

[8] C. Müller-Schloer, H. Schmeck, and T. Ungerer, Eds., *Organic Computing - A Paradigm Shift for Complex Systems*. Springer, 2011.

[9] M. Esteva, J. A. Rodriguez-Aguilar, J. L. Arcos, C. Sierra, and P. Garcia, "Institutionalising open multi-agent systems. a formal approach," Artificial Intelligence Research Institute. Spanish Council for Scientific Research. IIIA Research Report 2000-01, Tech. Rep., 2000. [Online]. Available: http://www.iiia.csic.es/Publications/Reports/2000

[10] S. M. Lee, R. Remington, and U. Ravinder, "A framework for modeling and simulating human behavior in complex systems," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, oct. 2005, pp. 3161 – 3166 Vol. 4.

[11] T. Kleinschmidt, X. Guo, W. Ma, and P. Yarlagadda, "Including airport duty-free shopping in arriving passenger simulation and the opportunities this presents," in *Proceedings of the 2011 Winter Simulation Conference (WSC)*, dec. 2011, pp. 210 –221.

[12] A. Verbraeck and E. Valentin, "Simulation building blocks for airport terminal modeling," in *Proceedings of the Winter Simulation Conference, 2002.*, vol. 2, dec. 2002, pp. 1199 – 1206 vol.2.

[13] Y. Li, "Static and dynamic complexity analysis of urban public transportation network: A case in shanghai," in *International Conference on Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007.*, sept. 2007, pp. 6382 –6385.

[14] V. Cheng, "Collaborative automation systems for enhancing airport surface traffic efficiency and safety," in *The 21st Digital Avionics Systems Conference, 2002. Proceedings.*, 2002.

[15] ——, "Evaluation plan for system-wide benefits of an airport surface-operation automation concept," in *The 23rd Digital Avionics Systems Conference, 2004. DASC 04.*, 2004.

[16] Y. Ju, J. Yu, A. Wang, and H. Che, "Research on simulation of airport fire emergency rescue based on swarm," in *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, june 2008, pp. 9022 –9027.

[17] M. Maier, "Architecting principles for system of systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[18] M. Fiosins, J. Fiosina, J. P. Müller, and J. Görmer, "Agent-based integrated decision making for autonomous vehicles in urban traffic," in *Advances on Practical Applications of Agents and Multiagent Systems*, ser. Advances in Intelligent and Soft Computing, Y. Demazeau, M. Pechoucek, J. Corchado, and J. Prez, Eds. Springer Berlin / Heidelberg, 2011, vol. 88, pp. 173–178.