IADIS International Journal on Computer Science and Information Systems
Vol. 9, No. 2, pp. 38-52
ISSN: 1646-3692

SCENARIO ASSISTANT FOR COMPLEX SYSTEM CONFIGURATIONS

Alexander Streicher. *Fraunhofer IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany.*

Daniel Szentes. *Fraunhofer IOSB.*

Wolfgang Roller. *Fraunhofer IOSB.*

ABSTRACT

We describe our concept and implementation of a mobile scenario assistant which helps with the automatic configuration of complex systems for demonstration scenarios. Presenting features of complex systems often requires profound expert knowledge which is more than often not available when needed ad-hoc on the spot. A solution approach is to create a machine readable model which can be used by assistance systems to help with the complicated setup and operation. The contribution of this work is the concept and implementation details of the mobile, distributed scenario assistant SCENAS. It is an assistance system to automatically configure components of a complex, heterogeneous system for its use in training and demonstration scenarios. The presented modeling approach of complex systems comprises state machine and standardized data flow principles with focus on feasible adaptability. Mobile devices are used as graphical user interfaces for "scenario remote control" and as mobile information brokers. The application is shown in a complex, heterogeneous system for image exploitation.

KEYWORDS

Assistance system, mobile learning, modeling and simulation, human computer interfaces

1. INTRODUCTION

The topic of this work is to provide operators and users of complex, heterogeneous systems with (mobile) assistance systems which hide the complexity of the configuration and technical aspects and show the users an abstract, non-technical view of the scenario and system configuration. Such challenges can be found where complex technical systems operate in the background, like in home automation, smart environments or medicine (Boswarthick et al. 2012; Cook & Das 2005; Schobel et al. 2013). Typically in home automation systems one does not want to know the technical aspects of the integrated sensors, controllers and actors, but to just have a remote control which takes care of opening up jealousies, the garage or regulate the temperature. In medicine the doctors and nurses do not want and do not have the time to adjust medical equipment but they just want to use it to help. One common solution is to insert an additional layer which hides the complexity from the users (Wiederhold 1992; Yilmaz & Ören 2009).

We present an approach of a mobile assistance and e-learning system called Scenario Assistant (SCENAS) which hides the complexity of the underlying configuration of a complex, heterogeneous system for image exploitation. The setup and operation of complex systems in demonstration scenarios require profound understanding of the internals of the involved components. One has to know the role and activities a specific system should play, as well as how the data and information flow is. SCENAS presents the user building blocks for scenario construction. In combination with a simulation environment the user can learn and train the handling of such systems. E-Learning and assistance systems can help experts as well as non-professionals to effectively setup complex systems for their utilization in demonstration scenarios. Assistance systems can abstract complicated workflows and take over an active role in setting up the target systems – like some kind of 'remote control'. Instead of manually configuring each and every system the configuration can be automatized. And when used in e-learning and training these assistance systems often give access to help- and learning material. Training environments can help amateurs as well as experts trying to understand such complicated systems. In combination with a simulation environment this opens up the possibility for training. The users can construct new scenarios using the building blocks and test them in virtual environments. In addition game elements will motivate the learners to actually play SCENAS to get a grasp of the interplay of the various complex systems.

The research question is how to optimally assist and help the user when planning and setting up demonstration scenarios for a complex, heterogeneous network of interconnected system appliances. Additional complexity is added by the experimental nature of such systems. Because of the experimental nature they tend to not being fully documented – this is often the case when they are part of active research projects. However, they are more than often actually used in real-life operation and actively used for field tests and demonstration scenarios. But still, experts have to setup and operate the various components. They have to configure each system to fit to a certain target scenario. Additionally those people are often specialists just for their own systems. Therefore in a heterogeneous network of multiple, interconnected systems many specialists are needed. However in field tests they need knowledge about the other systems as well. Because of complex internal structures for a non-professional this is almost always impossible.

One application of SCENAS is an experimental image exploitation system (ExBA). ExBA consists of different, complex systems, which offer technologies and support in solving different tasks and problems in the area of exploitation (Segor et al. 2011). Image exploitation includes automatic and interactive extraction of information from images and video recordings as well as its preparation and processing. Because the ExBa image exploitation system consists of heterogeneous, interconnected components with varying properties as well as varying technologies expert knowledge is required for their configuration and putting into operation. SCENAS can help here.

This extended version of (Streicher et al. 2014) is organized as follows: the next section on related work presents research and applications similar to ours. The following section on the system environment introduces the ecosystem of our mobile scenario assistant SCENAS and describes the architecture modeling approach. After that SCENAS and its components are presented. A short example of the application in a heterogeneous image exploitation network is then shown. As a last point we give concluding remarks and present future ideas.

2. RELATED WORK

Other work involved with scenario assistance systems can be found in the area of home automation or smart environments. This broad area includes computational architecture design, network protocols, intelligent sensor networks, action identification and prediction as well as philosophical considerations (Cook & Das 2005). The technological basis for the automated information exchange and system interoperability can be found in the machine-to-machine (M2M) technologies (Boswarthick et al. 2012). M2M by itself means by definition complex systems. However, a desirable path would be to offer plug-and-play installation and usage of such systems, in particular in respect to future easy to use home automation systems. This path is one of the main directions of this paper. An example for easy installation and usage in home automation is shown in (Abdulrazak & Helal 2006).

In this paper we used the Android mobile device framework. Mobile handhelds are commonly used to offer users ubiquitous access to information and communication protocols. Additionally they can be used as remote controls for computers and household and office appliances (Myers 2005; Myers et al. 2004). Manashty et al. present a smartphone-based remote control for controlling a smart-home environment (Manashty et al. 2010). Similar to the work presented by us they show how to integrate a mobile scenario assistant for home automation. Another example for mobile assistance systems for smart environments is the project MAIKE (Abdulrazak & Helal 2010). Similar to our approach MAIKE want ease the setup of heterogeneous device ensembles, in particular for smart, multimodal rooms. The users should not spend their time and efforts on complicated setup and operation requirements.

For distributed systems where various agents – or also named controllers – exchange data and information universally accepted engineering patterns exist, especially for modeling and simulation (Yilmaz & Ören 2009). The effective and interoperable modeling of heterogeneous systems can be achieved by aligning the architecture and interfaces to the Discrete Event System Specification (DEVS) (Zeigler et al. 2000). The presented work in this paper makes use of this principle. Other work also using DEVs for distributed systems and simulations is presented in (Mittal et al. 2009). Röhl et al. present a component framework that can be added as an additional layer on top of simulation systems to achieve a flexible composition of efficient simulation models (Röhl & Uhrmacher 2006).

So far, to the best of our knowledge, no mobile assistance system for scenario configuration applications has been presented yet. The application area of our mobile assistance system, *viz* image exploitation and surveillance, is an active research topic, especially the automatic surveillance and automation technologies (Bürkle et al. 2011; Heinze et al. 2010; Monari et al. 2008). The complexity of the developed systems and algorithms hinders non-experts to plainly use that technology. The presented mobile scenario assistant tries to shorten the gap between technology and its plain application.

3. SYSTEM ENVIRONMENT

The scenario assistant is embedded in a complex interconnected system for image exploitation. Reconnaissance and surveillance in civilian and military areas is often carried out in a network of complex heterogeneous subsystems. Image exploitation includes automatic and interactive extraction of information from images and video recordings as well as its preparation and processing.

3.1 Experimental System for Image Exploitation (ExBA)

One example of such an interconnected system is the Experimental System for Image Exploitation (ExBA) (Segor et al. 2011). Civilian and military personnel use it in training and demonstration scenarios in the application field of image exploitation, reconnaissance and surveillance. ExBA consists of various user interfaces, sensor platforms and processing algorithms (examples shown in Figure 1 and Figure 2).

Figure 1. Some components of the experimental system for image exploitation ExBA, (from left to right) digital sitation desk, stationary dome camera, land robot, quadrocopter drone

For example a digital situation desk offers multiuser touch displays both for mission planning and monitoring (Segor et al. 2011). The composite sensor network AMFIS (Bürkle et al. 2011) includes a variety of sensor platforms and networks, e.g. unmanned aerial camera platforms as quadro-copters or balloons, unmanned land-based vehicles, stationary cameras and unmanned underwater vehicles like mini-submarines. Another subsystem is the Coalition Shared Data (CSD) Server, based on a distributed database concept where all mission planning data and collected sensor data (pictures, video streams etc.) are stored. Also included is the ABUL subsystem which provides computer vision algorithms to create for example fine-tuned geo-location referenced image mosaics or super-resolution images (Heinze et al. 2010). To complete the picture a mobile assistance subsystem offers the user context-relevant help and learning material (Streicher et al. 2011).

The configuration of all these subsystems is complicated, time consuming and needs expert knowledge. Because of its experimental nature some parts are often subject to change and thus are not fully documented. This yields a complex environment where solutions with rapid adaptability possibilities are needed.

Thus to ease the configuration process and enable non-professionals to set-up the whole system for training and demonstrations the SCENAS scenario assistant has been developed. It is integrated on the XMPP-based communication layer which runs through the whole interconnected network.

a) Various sensor platforms b) Station/mission control c) Application scenarios

Figure 2. Example components of the interconnected system network ExBA. a) Left: various sensor platforms like quadrocopters, land robots as well as stationary and mobile cameras. b) Center: various control stations. c) Right: multiple application scenarios, like supervision of public festivals, image mosaicing and change detection for property surveillance or automatic 3D terrain generation

3.2 Modularized Scenario Description

In this work the term scenario means a planned series of system states and transitions. This is highly related to state machines. For the automatic scenario analysis and generation SCENAS needs a formalized, machine readable description of the scenario and its system environment. Different scenario types can be found in the literature. Typically scenarios are described as possible states of the world that represent alternative plausible conditions under different assumptions (Mahmoud et al. 2009). For an illustration of this concept compare Figure 3; all possible scenario paths are encompassed in a scenario funnel. Our scenario model is similar. Different possible outcomes mean that different states and transitions can be followed. However, the series of events follows a set of rules (analogous to the funnel). Each scenario consists of different sequences of system states. Each state can only be connected to compatible inputs and outputs of other system states (Figure 3).

To abstract and modularize the functions of complex interconnected systems standard information process paradigms and data flow standards can be applied, i.e. algorithms process some input and yield an output. To achieve interoperability in distributed systems the common principle is to standardize interfaces as well as input and output messages. Further standardization can be achieved by applying commonly accepted standards like the Discrete Event System Specification (DEVS) or the IEEE simulation standard High Level Architecture (HLA) (IEEE 2000; Zeigler et al. 2000). DEVS is a formalism for modeling and analysis of discrete event systems.

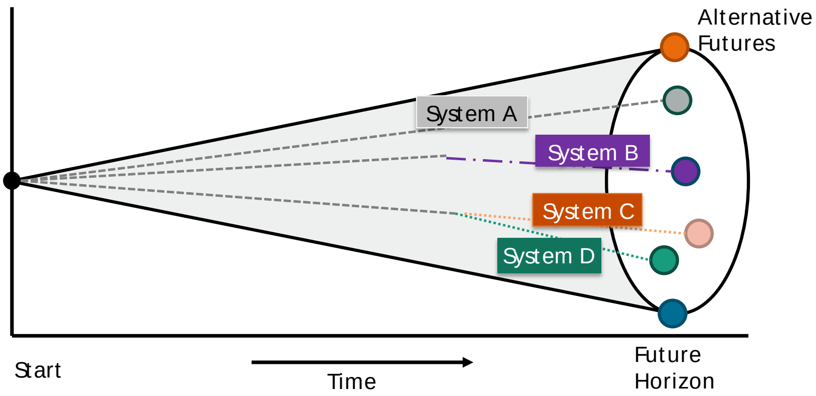

Figure 3. Conceptual diagram of a scenario funnel. Various systems are part of a scenario; multiple events are possible resulting in multiple alternative futures (outcomes). Adapted from (Timpe & Scheeper 2003)

For the scenario configuration assistant and our application in the image exploitation system ExBA we assume that the underlying processes can be described by state transition tables and has discrete events, like "object spotted" or "mapping process finished".

One possibility to achieve interoperability and abstract a complex system with multiple components is to identify the functions and to describe those functions as atomic functional elements (FE). Each functional element has at least an input and output edge, and each input and output is connected to other FEs. Hence inputs and outputs represent component borders. These borders can be used to define the level of semantic granularity of each functional element. Obviously, too much granularity would result in powerful yet vastly complex models, whereas too little granularity would not capture some important aspects of the system. Thus component borders hint at the level where to start and stop modeling functions as functional elements.

By following standardized schematics for input and output connections the functional elements are interchangeable and possible pathways can be found automatically. For each functional element exist multiple possibilities for input and output nodes. In a typical state machine model a source and a target state exist, i.e. start and end. In the graph representation of all valid combinations of functional elements multiple valid scenario paths can be found from start to end (Figure 4). Hence a scenario path is a valid sequence of processing instructions in the interconnected system. This can be seen as one possible scenario. With this formalization of a complex interconnected system a modular scenario description can be achieved. Additional interoperability to external systems can be achieved by following the DEVS schematics. The functional elements are similar to the atomic models in DEVS, in combination with the input and output port specifications they represent coupled DEVS models.

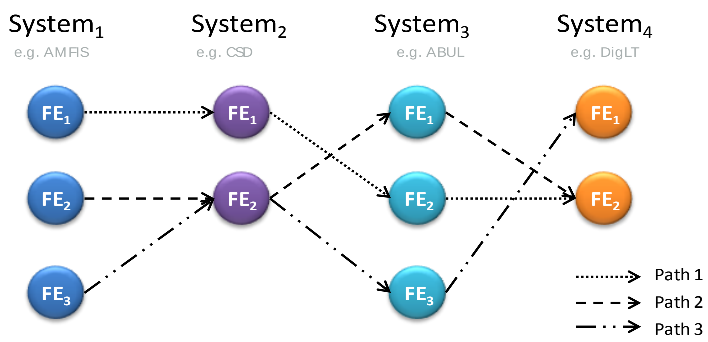

Figure 4. Principle of scenario paths with interchangeable functional elements (FE)

After all function elements (FE) have been defined the collection of states and transitions can be expressed as a directed acyclic graph (Figure 4, Figure 5). Every path from the source to the sink represents a valid scenario path. Branches exist because FEs can have multiple inputs and outputs (branching factor is greater or equal one). This graph can be used to search for scenarios with certain, user defined properties. To get all possible scenarios which include certain systems the graph has to be traversed where some FE nodes are set as active. Those paths, which do not have activated FE nodes, are dropped. An example for our system environment would be to find only those ExBA scenarios for which functions for quadrocopter flights (input FE), automatic image mosaicing (processing FE) and display on the digital situation desk (output FE) are included. Of course, these functions have processing steps before and after them, and they have to be in order. By following the graph paths the system knows that order and the pre- and postconditions. Depending on the branching factor a multitude of possible scenario paths exists. A limited number of simplified textual scenario path representations can then be presented to the user.

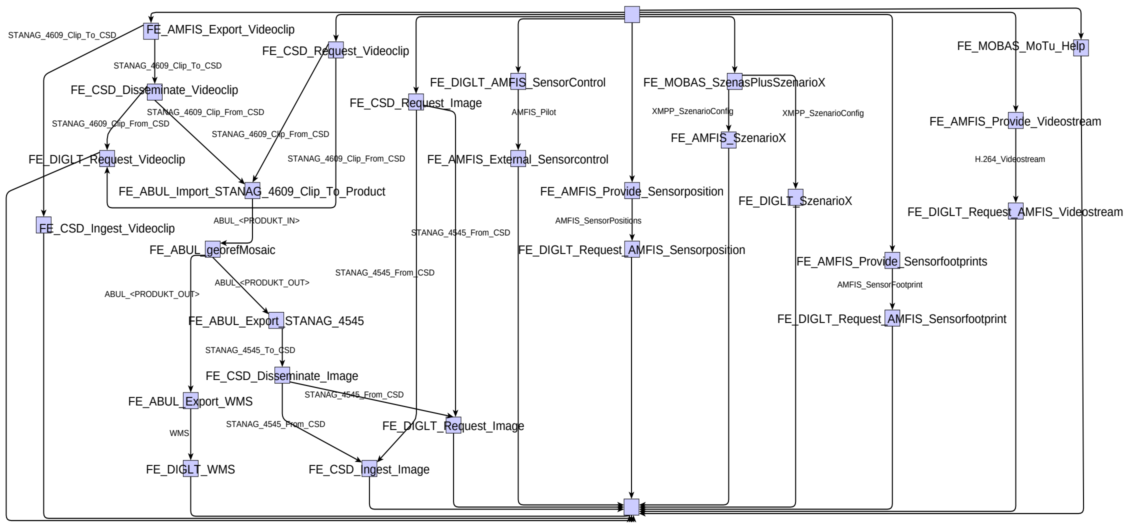

Figure 5. Partial acyclic directed graph of possible paths for the ExBA system automatically derived from the modularized system description with functional elements (FE)

4. SCENAS

The scenario assistant SCENAS consists of agents and viewing clients. The former are controller modules which are deployed on each appliance system, the latter contain the GUI and is implemented as a Web-based mobile application.

4.1 Architecture

The whole system is a distributed system with server and clients. Basically it follows the standard client-server architecture model. The clients connect to managing server hubs to interchange messages with the modules and agents in the back. A key aspect of the whole architecture is the communication layer based on the eXtensible Messaging and Presence Protocol (XMPP) (Saint-Andre et al. 2009). XMPP is used here as an extensible Message Oriented Middleware (xMOM) platform for machine-to-machine communication. From a technical perspective, by using XMPP on the communication layer the underlying complexity of message handling can be greatly reduced. Alternatively one could use competing communication protocols like Bonjour or the Service Location Protocol (SLP). All the managing aspects of distributed message communication are done by this standardized communication protocol and associated middleware. Since XMPP offers the possibility to have multiple interconnected XMPP servers, this architecture is open for scalability, i.e. decentralized locations for different scenario systems. The message payloads are encoded as XML; in XMPP called XML stanzas. An XML stanza is a discrete semantic unit of structured information (Saint-Andre et al. 2009). They are sent between the communicating agents and servers (Figure 6). Scenario configuration messages are of the XML stanza type <*message/*>, whereas the connected system components register themselves in the XMPP network via the stanza type <*presence/*>.

The reason for a decentralized design where the configuration parameters are stored locally on the devices was to achieve flexibility. The users should be able to adjust, i.e. reconfigure, or create new scenarios. This can be easily achieved by extending the XML schema with new semantic elements or by just using a key-value-pair semantic (Figure 7).

An additional advantage of using XMPP as communication layer is that one can easily monitor the message bus via a variety of already existing XMPP clients. Since the communication takes place in a normal XMPP multiuser chat room every authorized XMPP client can join that chat room and observe the incoming SCENAS messages.

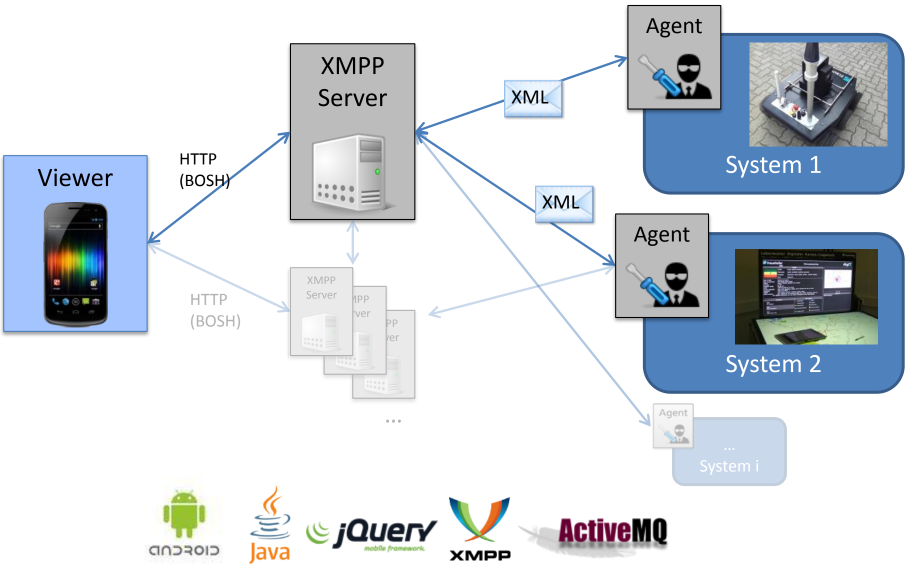

Figure 6. System architecture with XMPP servers, handheld clients and target systems with associated agents

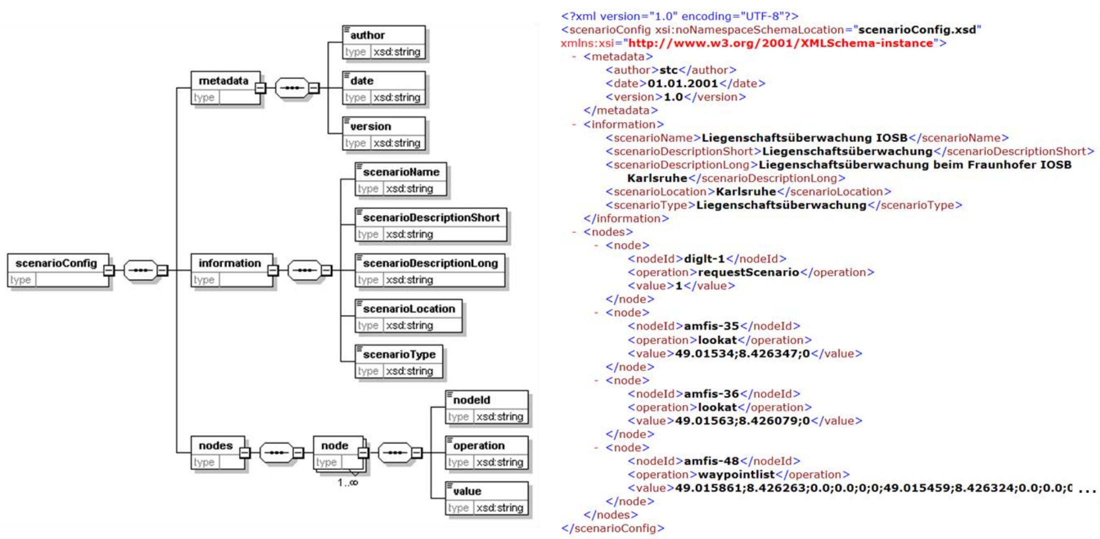

Figure 7. XML schema (left) of scenario configuration message and example (right)

4.2 SCENAS Agents and Controllers

To control the actual appliances we implemented agents – also often called adaptor, proxy or bridge – which translate the SCENAS commands to the specific target system. The SCENAS agent utilizes a plug-in concept to reduce overall complexity by encapsulating system-specific logic in own modules.

For each involved system, which should be automatically configured, a module has to be created. Each module contains transformation rules to translate the SCENAS scenario configuration messages to the specific language of the target system. It acts like a mediator between the SCENAS internal representation of scenario parameters and the specific interfaces of the target systems.

Each SCENAS Agent can be implemented in an omnipotent way, i.e. it carries all modules for every involved system. Unused modules can be disabled. This concept makes it easy to deploy the SCENAS agents on the target systems because little adjustments have to be made. Activated modules register themselves at the XMPP network in a dedicated multiuser chat room. The presence of a certain system in the chat room is an indicator that this system is online and ready; this status can then be presented to the user in the SCENAS Viewer.

Figure 8. SCENAS graphical user interface (GUI; in German). Catalogue of scenarios, checklist

4.3 SCENAS Viewer

The Android application SCENAS Viewer is the graphical user interface of the SCENAS system. It provides the user with an intuitive view of the available scenarios, gives status information of system availability and interfaces to start a scenario.

Following the model-view-controller principle (MVC) the SCENAS Client is realized as a web-based application. The implementation of the presentation layer is done in HTML5+CSS3, the control logic is done in JavaScript, and the model uses XML as data representation. The controller makes use of the well-known JavaScript library jQuery and jQuery mobile. The JavaScript library *Strophe.js* is used as XMPP library to communicate with the XMPP network. Since vanilla JavaScript does not allow persistent TCP/IP connections the XMPP extension Bidirectional-streams Over Synchronous HTTP (BOSH) has to be used. BOSH is a transport protocol which allows a bidirectional stream between two

XMPP entities. It uses the long polling technique to keep up the XMPP connection (Saint-Andre et al. 2009). As the client is realized as a platform independent web-based application SCENAS Client can be started on mobile devices with Android or iPhone as well as desktop computers. An example of the graphical user interface is shown in Figure 8. A list of available scenarios is shown on the left; the screen in the center gives a detailed description of the scenario and an overall status report of all involved systems (e.g. "3 out of 6 are ready"); the right screen gives a status report on each participating system with traffic light colors indicating either ready (green) or not-ready (red).

4.4 Workflow

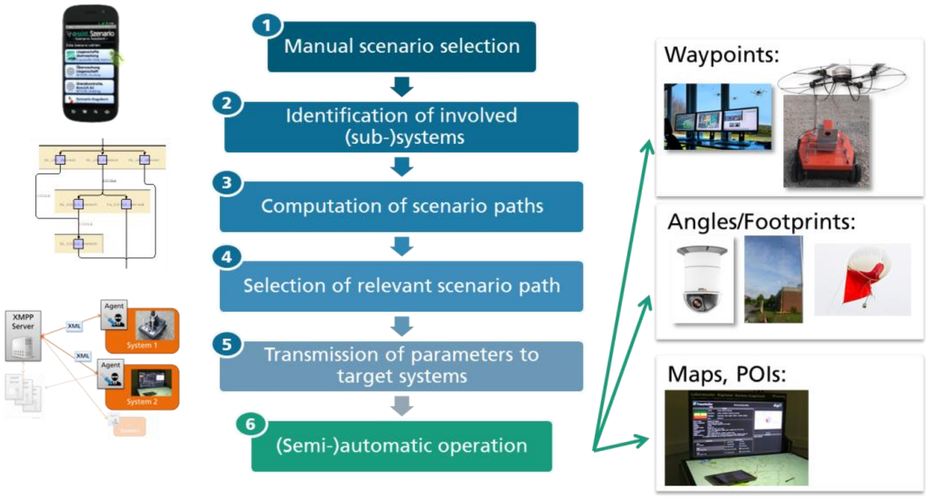

Figure 9. Typical SCENAS workflow. Scenarios are selected by the users, the system determines involved systems and computes possible scenario paths and transmits the configuration parameters

A typical SCENAS workflow consists of six main steps, as can be seen in Figure 9:

1) The user manually selects a possible scenario (*cf.* Figure 8, left).
2) SCENAS automatically identifies which (sub-)systems are involved in the chosen scenario and
3) computes all possible scenario paths (*cf.* scenario paths in Figure 5).
4) 4) Depending on user preferences the possible paths are filtered and the ones relevant for the user are automatically selected.
5) SCENAS loads the scenario configuration parameters (*cf.* Figure 7) and transmits them to the SCENAS server which distributes the parameters to the target systems (*cf.* distributed architecture in Figure 6).
6) The target systems start to operate according to the scenario configuration parameters.

Appliances which are safe to operate on themselves can be controlled fully automatic. Appliances with safety concerns are controlled in a semi-automatic way. For example, the change of perspective of the stationary dome cameras is executed fully automatically; the flights of quadrocopters along transmitted waypoint lists must be pre-checked by operators;

though technically possible the land robots in fact do not drive by themselves because of safety concerns, but instead are steered by pilots along waypoint lists sent by SCENAS.

5. APPLICATION

SCENAS has been tested in the heterogeneous image exploitation network ExBa (Bürkle et al. 2011; Segor et al. 2011) (cf. section 41). In the following an example of a demonstration scenario is outlined and the outcome of the application of SCENAS is explained.

One possible demonstration scenario is to survey a defined area. The ExBa components work together and continuously interchange data and information. Human operators plan and analyze the information to decide which actions to take e.g. raise an intruder alert or explore unobserved areas.

The scenario is as following: a system manager decides on starting a scenario “surveillance of area *X*”. The digital observation table DigLT should load a schematic map of area *X*, AMFIS stationary cameras *A* and *B* should point to spots *X.a* and *X.b*. The waypoint list for area *X* and a specific route are loaded into the pilot interface of the AMFIS land robot (Figure 10).

When a user selects this demonstration scenario the mobile client sends the specific configuration parameters for this scenario via XMPP to the agents on the appliances. The agents then communicate with the controllers of the target devices and translate the incoming configuration messages. For example, the dome cameras *A* and *B* should point to *X.a* and *X.b*. The agents receive new coordinates and translate them to new angular parameters for the cameras. As an application example for this camera scenario see Figure 11.

The whole message communication can be easily observed by connecting to the XMPP server and the dedicated chat room with a common XMPP client. In this small example, which is part of a bigger scenario, the dome cameras changed their viewing angle according to the exchanged XML-based configuration messages. An important aspect is that the controlling of the various target systems, i.e. here the motors of dome cameras, is still done by their native software components or by the specific system’s middleware – and not by the introduced agent modules. The agents only act as an adaptor (proxy) to the actual controller. In a system environment similar to ExBA one only has to implement agent modules for the first hierarchy. In ExBA we implemented agent modules for the composite sensor network AMFIS and the digital situation desk DigLT.

Figure 10. Automatic transfer of new waypoint list to pilot interface of land robot

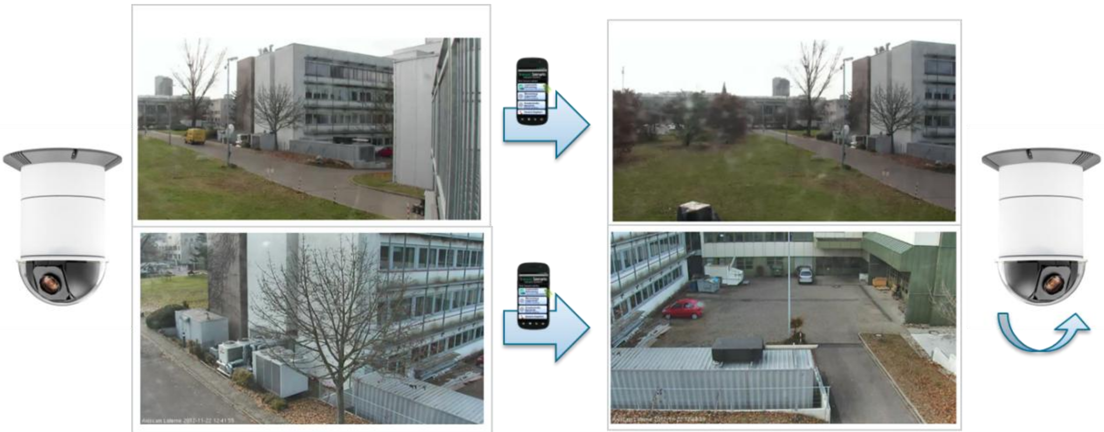

Figure 11. Application of different scenario parameters to the stationary cameras

The scenario configurations are based on the formalized subsystem functional descriptions created beforehand by the system designers. Of course the first step is to identify which system components should – and actually can – be configured in an automatic way (atomic models). These subcomponents are then described functionality-wise by the introduced functional element (FE) principle (section 3.2). In ExBA experts from the various subsystems described the functionalities of their subsystem by an agreed-on semantic formularization specification which is similar to the DEVS schematics. In brief, all sets of possible input and output events are identified and described with unique and aligned port identifiers. The coupled models then represent possible scenario event stations which can be followed algorithmically to determine all possible scenario paths (cf. section 3).

6. CONCLUSION

We presented the mobile scenario assistant SCENAS and its distributed architecture. SCENAS is a scenario assistance system to automatically configure components of a complex, heterogeneous system for its use in training and demonstration scenarios. The graphical user frontend is primarily web-based and deployed on mobile devices; the backend is based on a distributed architecture and the agents communicate via standard XMPP. SCENAS has been successfully applied in a heterogeneous interconnected system for image exploitation. Various user interfaces, sensor platforms and processing algorithms can automatically be configured. Software agents play the role of adaptors and non-intrusively translate and transfer configuration parameters to the target systems.

Future research will take the scenario assistant one step further to offer an authoring utility to the users so they can by themselves add and edit scenario configurations. This is based on the visual programming environment principle. Basis for this must be a standardized, machine-readable description of the involved systems and their functionalities. The idea is to provide the users with a puzzle-like graphical user interface with templates so they can patch together new functional combinations in a standardized fashion.

REFERENCES

Abdulrazak, B. & Helal, A., 2006. Enabling a Plug-and-play integration of smart environments. *2006 2nd International Conference on Information & Communication Technologies*, 1.

Abdulrazak, B. & Helal, A., 2010. Enabling a Plug-and-play integration of smart environments. In *2006 2nd International Conference on Information & Communication Technologies*. Wismar; Germany: IEEE, pp. 820–825.

Boswarthick, D., Elloumi, O. & Hersent, O., 2012. *M2M Communications: A Systems Approach*,

Bürkle, B. et al., 2011. Autonomous Geo-referenced Aerial Reconnaissance for Instantaneous Applications. *International Journal On Advances in Systems and Measurements SYSMEA 2011*, 4(3), pp.192–202.

Cook, D. & Das, S., 2005. *Smart environments: Technology, protocols and applications*, Hoboken; N.J: Wiley.

Heinze, N. et al., 2010. Image exploitation algorithms for reconnaissance and surveillance with UAV D. J. Henry, ed. *Proc. SPIE, Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications*, 7668(VII), p.76680U–76680U–11.

IEEE, 2000. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. *IEEE Std. 1516-2000*, pp.i –22.

Mahmoud, M. et al., 2009. A formal framework for scenario development in support of environmental decision-making. *Environmental Modelling and Software*, 24, pp.798–808.

Manashty, A.R.A., Rajabzadeh, A. & Jahromi, Z.Z.F., 2010. A Scenario-Based Mobile Application for Robot-Assisted Smart Digital Homes. *IJCSIS*, abs/1009.5(5).

Mittal, S. et al., 2009. DEVS/SOA: A Cross-Platform Framework for Net-centric Modeling and Simulation in DEVS Unified Process. *SIMULATION*, 85(7), pp.419–450.

Monari, E., Voth, S. & Kroschel, K., 2008. An object- and task-oriented architecture for automated video surveillance in distributed sensor networks. In *Proceedings - IEEE 5th International Conference on Advanced Video and Signal Based Surveillance, AVSS 2008*. pp. 339–346.

Myers, B. a., 2005. Using handhelds for wireless remote control of PCs and appliances. *Interacting with Computers*, 17(3), pp.251–264.

Myers, B.A. et al., 2004. Taking handheld devices to the next level. *Computer*, 37(12), pp.36–43.

Röhl, M. & Uhrmacher, A.M., 2006. Composing simulations from XML-specified model components. *Proceedings of the 38th conference on Winter simulation*, pp.1083-1090.

Saint-Andre, P., Smith, K. & TronCon, R., 2009. *XMPP: The Definitive Guide: Building Real-Time Applications with Jabber Technologies* 1st ed. O. Media, ed., O'Reilly Media.

Schobel, J. et al., 2013. Using Vital Sensors in Mobile Healthcare Business Applications. *9th Int'l Conference on Web Information Systems and Technologies (WEBIST 2013), Special Session on Business Apps*, pp.509–518.

Segor, F. et al., 2011. Instantaneous Autonomous Aerial Reconnaissance for Civil Applications. *International Academy, Research, and Industry Association -IARIA-: GlobeNet 2011 : International Conference on Systems (ICONS)*, pp.72–76.

Streicher, A. et al., 2011. Integrierte Mobile Learning Lösung für ein heterogenes Roboterverbundsystem in der Bildauswertung S. Friedrich, A. Kienle, & H. Rohland, eds. *DeLFI 2011: Die 9. e-Learning Fachtagung Informatik - Poster, Workshops, Kurzbeiträge, Technische Universität Dresden, Germany, 5.-8. September 2011*.

Streicher, A., Szentes, D. & Roller, W., 2014. SCENAS - Mobile Scenario Assistant for Complex System Configurations. In A. P. Abraham, A. P. dos Reis, & J. Roth, eds. *International Conference on Theory and Practice in Modern Computing, MCCSIS 2014*. Lisbon, Portugal: IADIS, pp. 157–165.

Timpe, C. & Scheeper, M.J., 2003. A look into the future: scenarios for distributed generation in Europe. SUSTELNET. Policy and Regulatory Roadmaps for the Integration of Distributed Generation and the Development of Sustainable Electricity Networks, p.25.

Wiederhold, G., 1992. Mediators in the architecture of future information systems. *Computer*, 25(3), pp.38–49.

Yilmaz, L. & Ören, T., 2009. *Agent-Directed Simulation and Systems Engineering*, Wiley.

Zeigler, B.P., Kim, T.G. & Praehofer, H., 2000. *Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems* 2nd ed., San Diego; Calif. [u.a.]: Academic Press.