

Connecting Technical and Non-Technical Views of System Architectures

Li Li¹

¹Embedded Systems Institute
Eindhoven, the Netherlands
e-mail: li.li@esi.nl

Jozef Hooman^{1,2}

²Computing Science Department
Radboud University
Nijmegen, the Netherlands
e-mail: jozef.hooman@esi.nl

Jeroen Voeten^{1,3}

³Department of Electrical
Engineering
Eindhoven University of
Technology
Eindhoven, the Netherlands
e-mail: J.P.M.Voeten@tue.nl

Abstract—When designing, adapting or extending a complex cyber physical system, it extremely difficult to guarantee that all parts cooperate correctly, especially when many parties are involved in the development process. We propose a new approach to identify and to solve problems in the early development phases, thus reducing costs in the system implementation, test and integration phases. This is achieved by making a global, holistic model of the system with different views. From the technical point of view, a system simulation model is constructed which can be used to validate whether the global system architecture satisfies the main requirements. To bridge the gap between technical specialists and non-technical stakeholders, this technical system view can be connected to a user-friendly animation tool. This allows the creation of an interactive prototype with different stakeholder views. The approach will be illustrated by a case study in the healthcare domain.

Keywords—cyber physical systems; system architecture; architectural views; modeling; simulation; animation; healthcare

I. INTRODUCTION

In many domains, existing systems are adapted and extended, for instance, to increase societal benefits in a time-efficient and cost-efficient way. This includes new functionality, combinations of information from different sources, and new connections between devices. For example, the integration of current traffic management systems with next-generation automotive safety technologies could dramatically decrease the number of victims due to traffic accidents. Also in the healthcare domain innovations will have a large impact, due to the aging population; in 2045 the number of people older than 60 years will have grown from 600 million to 2 billion [1]. How to increase quality and productivity in the healthcare care cycle, that is, from early diagnosis to treatment and monitoring, is a challenging topic. As mentioned in [12], an important factor toward efficiency and cost-effectiveness in healthcare is an improvement of the ability to share information across systems and between care organizations. Then, new technologies from different areas of healthcare can be applied and integrated in current medical products and systems. For instance, novel medical image analysis methods can be used for different imaging modalities, such as X-ray, computed tomography (CT) and magnetic resonance imaging (MRI). More patient-specific

decision support systems together with different medical knowledge sources could be connected to the existing hospital information systems. These result in complex cyber-physical systems, where several devices, information sources and humans work together to obtain an improved service level.

The complexity of such heterogeneous systems makes it extremely difficult to develop them. Especially, since many parties are involved in the developing process, it is very difficult to guarantee that all parts cooperate correctly to deliver dependable services. Typically, many problems become visible only when parts of the system are integrated, resulting in long test and integration times. Problems such as interface mismatches and performance bottlenecks are usually detected rather late in the development process. In the healthcare domain, [13] identifies three core reasons of many failures: (1) integration and standardization; (2) human-computer interaction and the structure of data/information/ knowledge; (3) socio-technical and organizational issues.

To cope with these problems, system-level design methods have been devised, including architecture description languages (ADLs). These methods focus on modeling the system and conducting, for instance, performance analysis before actual system implementation. A well-known architecture description language for system-level design is the Architecture Analysis and Design Language (AADL) [7], which initially was developed for the modeling and analysis of systems in the avionics domain [8]. Currently, it is widely used in other domains as well [9][10]. A general purpose modeling language for systems engineering applications is SysML (Systems Modeling Language) [11]. A common weakness of these architecture description languages is they describe a system mainly from a technical viewpoint which is often difficult to understand for non-technical stakeholders. Usually, however, the non-technical aspects play a very important role when adapting or extending a complex system. In [14] it has been observed that in industry mainly text documents are used, with relatively little attention for visualization in the early design phases. They note that visual notations such as SysML have a rather complicated syntax and are difficult to learn and to understand.

The aim of our approach is to discover and to solve problems in the earlier design stages to reduce costs and

effort in the system implementation, test and integration phases. This is done by making a global, holistic model of the system with views on different aspects. As defined, e.g., in [15], holistic approaches are those that consider systems in their entirety rather than just focusing on specific properties or specific components.

From the technical point of view, we built a system simulation model which can be used to validate whether the design of the global system architecture satisfies the main requirements. To obtain a user-central design point of view, an animation tool has been connected to this system simulation model. This tool makes it possible to construct different views for various stakeholders, such as architects, engineers, project managers and customers (e.g., patients and clinical physicians), to collect and to validate non-technical requirements of users. Furthermore, it can bridge the gap between technical viewpoints from technical specialists and non-technical aspects from other stakeholders. In this way, we obtain a user-friendly interaction between the simulation of the system architecture and a convenient representation of the architectural behavior for a particular stakeholder.

The structure of this paper is organized as follows. Section II introduces the case study that will be used to illustrate our approach. Section III describes the modeling and simulation language POOSL which formalizes the technical view on the global system architecture. In Section IV we present a new extension of this language with a socket mechanism such that it can be connected to interactive animations constructed with the Adobe Flash technology. This allows the construction of multiple views for non-technical stakeholders. Concluding remarks and future work can be found in Section V.

II. CASE STUDY

The work presented here has been conducted as part of the Care4Me (Cooperative Advanced REsearch for Medical Efficiency, www.care4me.eu/) project of ITEA2 (Information Technology for European Advancement) program. The project aims at increased quality and productivity in the care cycle by using more advanced medical image analysis techniques and by more patient-specific decision support systems. These are combined with different medical knowledge sources such as health records, medical evidence and medical image-based models.

As shown in Fig. 1, the Care4Me project concentrates on eight clinical use cases related to three different disease areas: neurodegenerative disease, cardiovascular disease and oncology. Innovations on these use cases are investigated by 25 partners from 5 different countries (Spain, Finland, France, Greece, and the Netherlands).

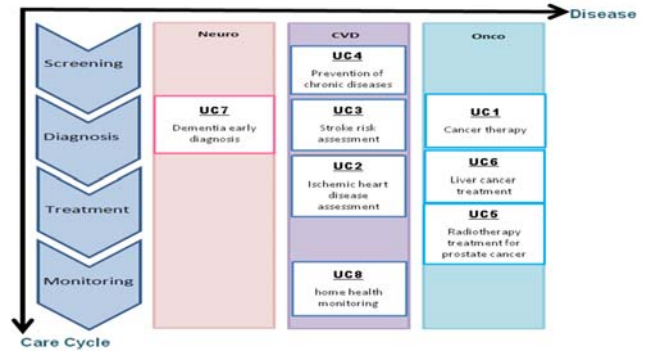


Figure 1. Eight use cases cover different diseases and care cycles.

The use cases cover not only the individual patient care cycle, ranging from screening, diagnosis to treatment planning and follow-up monitoring, but also include disease prevention focusing on larger populations with the objective of screening or risk group identification.

As an example, use case 2 about ischemic heart disease assessments is discussed in more detail. Ischemic heart disease can manifest itself as a slowly progressive process (plaque build-up in the coronaries causing stenosis) or in an acute phase (e.g., due to plaque rupture); both scenarios require different diagnostic pathways. Use case 2 describes both these diagnostic pathways. The interventional cardiologist needs state-of-art image quality and enhanced visualization for diagnosis (pre-interventional) and treatment (during an intervention) at minimal radiation dose (X-ray, CT) or scanning time (MR) in 2D or 3D images. In addition, better image quality improves real-time intervention and automatic optimization, quantification, diagnosis and decision support.

Fig. 2 shows the work flow from the patient point of view. A patient with heart complaints enters a hospital through a referral from a primary health care provider or an emergency unit. Pre-interventional diagnostic information is collected, possibly including imaging with different modalities, such as CT, MR, US or X-ray. These images are automatically optimized for subjective evaluation or automatic quantification and analysis. The images are stored in a PACS (Picture Archiving and Communication System) database. The images are reviewed by a cardiologist and optimized interactively or according to his pre-adjusted personal preferences. If necessary, an intervention will follow under X-ray guidance with real-time image optimization for subjective and automatic interpretation. Images are stored in a PACS database for reuse later on, e.g., in follow-up monitoring or to discuss results with colleagues and the patient.

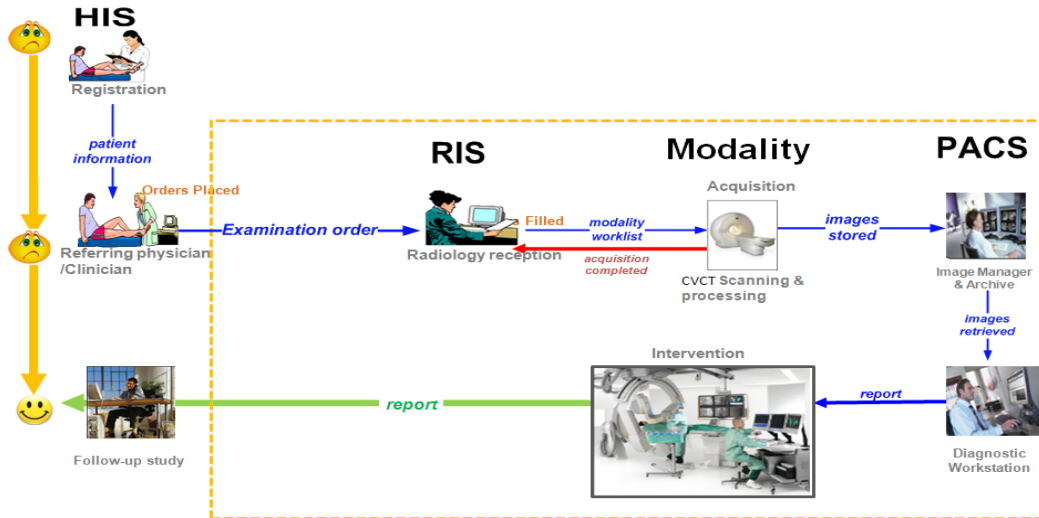


Figure 2. The work flow from the patient point of view in use case 2.

The designers of a system which should support use case 2 have to deal with different devices, medical knowledge sources and human-machine interactions (for example, among the referring physician, cardiologist, patient and machines or devices). In particular, it is very complex to coordinate the collaboration of different parts, such that they work together properly and improve efficiency and productivity of the health care cycle.

III. HOLISTIC SYSTEM MODEL

When designing, adapting or extending a complex system it is very useful to have a global holistic model. Especially during the early phases of the development process, such a model assists designers in evaluating design ideas and making reasonable design decisions. Moreover, it supports the analysis of system functionality and overall performance aspects before the actual implementation of the system.

To construct a global system model, we have used a general-purpose system-level design methodology called Software/Hardware Engineering (SHE) [2]. In this methodology, models are expressed in the Parallel Object-Oriented Specification Language (POOSL). A POOSL model consists of a number of process objects, which can be clustered in a hierarchical way. Processes communicate by means of synchronous communication via channels, similar to the message passing mechanism of CCS [6]. Timing information can be expressed by means of delay statements. In addition to processes, POOSL also contains data objects which are grouped in classes. POOSL has a formally defined semantics, based on a two-phase execution model which distinguishes local and communication actions, which do not consume time, and time transitions.

POOSL models can be constructed by means of the editor of the SHESim tool. Moreover, with this tool the

models can also be simulated. The tool supports both step-wise and continuous simulation. During simulation, the state of objects can be inspected and a so-called interaction diagram (similar to a message sequence diagram) shows the communication history. The SHESim tool is freely available [3].

We use the case study of Section II to illustrate the construction of a global system model using POOSL and SHESim. Our approach starts with an informal overview and structure of the whole system, including its users, as described in Section III.A. Next a formal system model is constructed in POOSL, as shown in Section I.A. Simulation and interaction diagrams are described in Section B.

A. Informal system overview

To get an overview of the healthcare applications developed in the Care4Me project, the global system architecture has been constructed based on all use cases and related user requirements. An outline of this architecture is shown in Fig. 3. It consists of four parts:

1. A user control part which includes all human factors involved in the Care4Me use cases, such as the patients, the clinic physician, and the radiologists.
2. A devices layer where all the modalities and devices used in this project have been gathered.
3. An information layer formed by all kinds of information systems related to all use cases, such as PHR (Personal Health Record) and PACS. Those information systems are passive process or represent databases which store information exchanged between different layers.
4. An applications layer consisting of innovative applications developed in the Care4Me project to support all use cases.

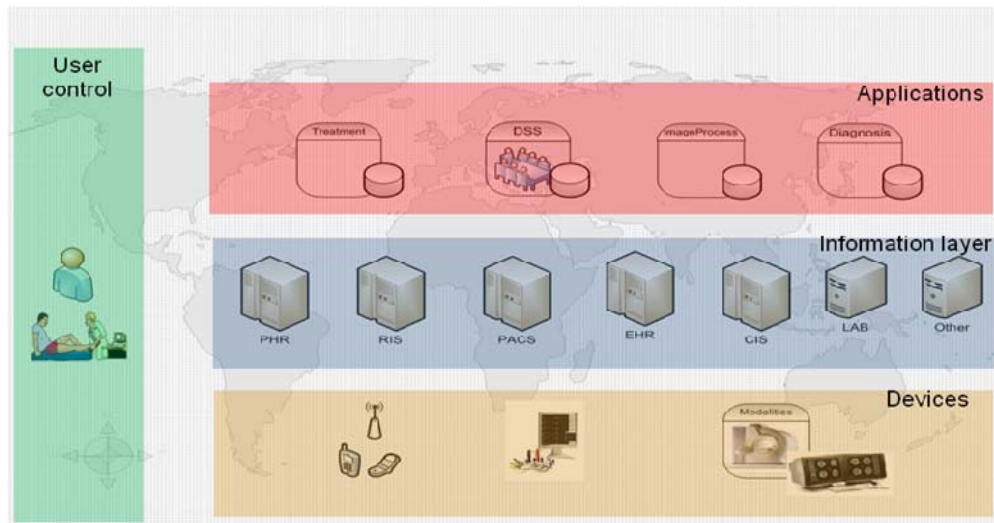


Figure 3. Global system architecture.

B. Formalising the global system view in POOSL

The global outline of the previous section is formalized in POOSL, using the same high-level structure, but adding much more detail to obtain an executable model. Hence, there are four parts in this POOSL model depicted in Fig. 4, corresponding to the four parts described in Figure 3. Each part has been modeled in detail by additional clusters and processes in POOSL.

The devices layer includes an ECG (ElectroCardioGram) process, which provides ECG diagrams, and a process Mobile which models information received via mobile phones (this is used in use case 8 on home health

monitoring). The Cluster Modalities includes all used imaging modalities such as CT, MR and X-ray. The Information layer consists of relevant information systems modeled by passive objects PHR (Personal Health Record), RIS (Radiology Information System, PACS and EHR (Electronic Health Record). The Cluster DecSource consists of sources that are used for medical decisions, such as CIS (Cardiology Information System) and LAB (Laboratory Information System). The application layer contains the innovative applications for all the use cases of the Care4Me project.

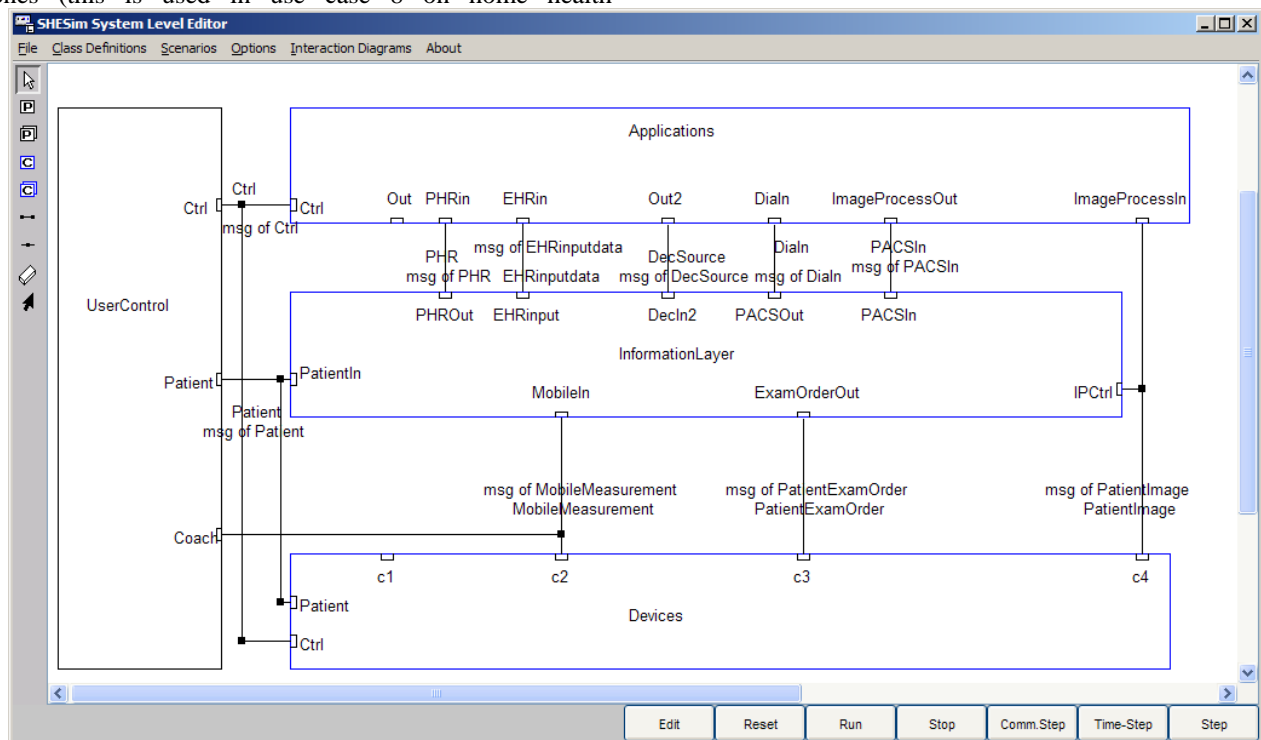


Figure 4. The global system architecture modeled in POOSL.

models with Flash animations, constructed with Adobe Flash Professional [4]. In this way, while an executable POOSL model is running, the synchronized Flash animation can provide vivid non-technical views for different stakeholders.

In Section A we design a Flash animation which will be connected to the POOSL model. To realize the connection, we extend POOSL with a new socket mechanism as described in Section IV.B. Section I.A presents the connection between POOSL and Flash and the application to our case study.

A. Design of a Flash animation

Adobe Flash Professional [4] is an easy-to-use visual multimedia tool. It can be used to quickly create interactive content and vivid animations. Hence, it has been chosen to connect with the POOSL model to implement interactive multiple animation views for non-technical stakeholders.

When connecting a Flash animation to the POOSL model, the Flash application is used as a socket client. In an Adobe Flash application, the Socket class can be used to accomplish this functionality [5], such as making a connection to a socket server (in our case the POOSL model) and exchanging messages with this server.

The design of the animation depends on the needs and requirements of the target users and stakeholders. As an example, the design for use case 2 is explained. While designing the animation for this use case, the scenario has been divided into the following steps:

1. Start: A patient feels heart pain and enters a hospital.
2. ECG: ECG has been used as an emergency check; the ECG result can be taken as the decision point: if the patient does not suffer from an urgent disease, monitoring can be conducted for this patient; if the ECG result shows a serious problem for this patient, then the patient needs further treatment immediately.
3. RIS and CT: The patient has to be sent to the radiology department, where the Radiology Information System (RIS) is involved, for pre-interventional diagnosis by using the modality CT.
4. Diagnosis: During the diagnosis by the CT modality, this patient's real-time images are automatically optimized and processed. After image processing, these images are reviewed by a cardiologist and other specialists.
5. Intervention: If necessary, an intervention will follow under X-ray guidance with real-time images. After the treatment, those images are stored for follow-up monitoring.
6. Monitoring: After the intervention, the status of the patient is supervised by clinicians and nurses.

C. Connecting POOSL and Flash

After extending POOSL with a socket mechanism and allowing a Flash animation to connect to a socket server in

The input interface of the Flash animation has been designed as shown in Fig. 7. The Connect button can be used to establish a socket connection with a socket server (with a certain server IP number and port number). After clicking the Connect button, it turns into "Connected" if a successful connection has been made. Based on the scenario steps of use case 2, six other buttons (Start, ECG, RIS & CT, Diagnosis, Intervention and Monitoring) have been designed to send the corresponding message to the server (the POOSL model in this case).

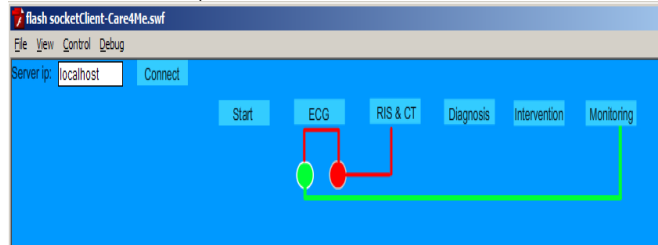


Figure 7. Design of animation input for use case 2.

The output part consists of two screens which display simultaneously two different views for non-technical stakeholders. The main animations are shown in Fig. 11 and explained in more detail later. The main idea is that the left one visualizes the work flow steps for the use case described above. The right screen is to show the dynamic transfer of information flow in the system architecture diagram; it visualizes the information flows in the system.

B. Including sockets in POOSL models

To allow POOSL models to communicate with different applications, such as Adobe Flash, the socket mechanism has been introduced by extending POOSL with a basic process class called Socket.

The mechanism of the socket connection between the POOSL model and other application, such as Adobe Flash, is depicted in Fig. 8. The Socket process in the POOSL model acts as a socket server. Once the socket connection is available, the Socket process can exchange messages with a client, such as socket client in Flash.

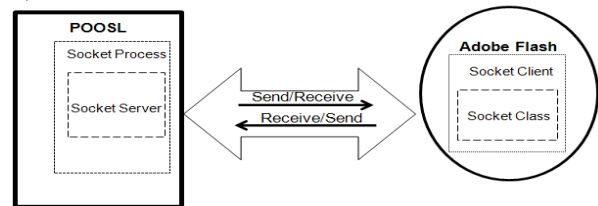


Figure 8. The mechanism of the socket connection between POOSL and Flash.

A Timer process is added to have some non-terminating activity in the model; otherwise the POOSL tool would detect termination or report a deadlock.

POOSL, the main question is which messages should be exchanged between the two models. It is important to realize that they have a different notion of time; Flash models are

executed in real-time whereas POOSL has a notion of simulation time. A delay of 5 seconds in POOSL might be much shorter in real-time and this could be very inconvenient for user interaction. Consequently, we have chosen to let the Flash animation control the POOSL model.

For our own validation of the message exchange between the Flash animation and the POOSL model, again interaction diagrams have been used to visualize the communication between the socket and the rest of the POOSL model.

For the validation of the model by non-technical stakeholders, now the Flash animation can be exploited. Fig. 9 shows the Flash animation screenshots for use case 2. After connecting with the executable POOSL model, the Flash animation visualizes the work flow steps on the lower left side and displays the information flow in the system architecture diagram on the lower right side:

- By clicking the Start button in Fig. 9a, the Flash animation has been programmed to send the message “Start” to the socket server in POOSL model which forwards this message to the UserControl process. This process triggers the architectural layers of the POOSL model (send the patient’s information to the ECG process). Meanwhile, the POOSL model sends the feedback to the Flash animation to start the first scenario step in Figure 9. a.
- After finishing the first scenario step, the user can click the ECG button to send message “ECG” to the POOSL model to perform the ECG check for the patient. Then, the POOSL sends the feedback back to the Flash animation to display the second scenario step in Fig. 9b. The ECG decision support process in the POOSL model is synchronized with the Flash user inputs. There are two options for the user to choose, namely, whether the ECG result shows an urgent disease or not. If the ECG result is not serious, monitoring can be conducted for this patient; if the ECG result is urgent, the patient needs RIS&CT, Diagnosis and Intervention as the treatment.
- The other scenario steps, such as RIS&CT in Fig. 9c, Diagnosis in Fig. 9d, Intervention in Fig. 9e and Monitoring in Fig. 9f have been implemented in the same way.

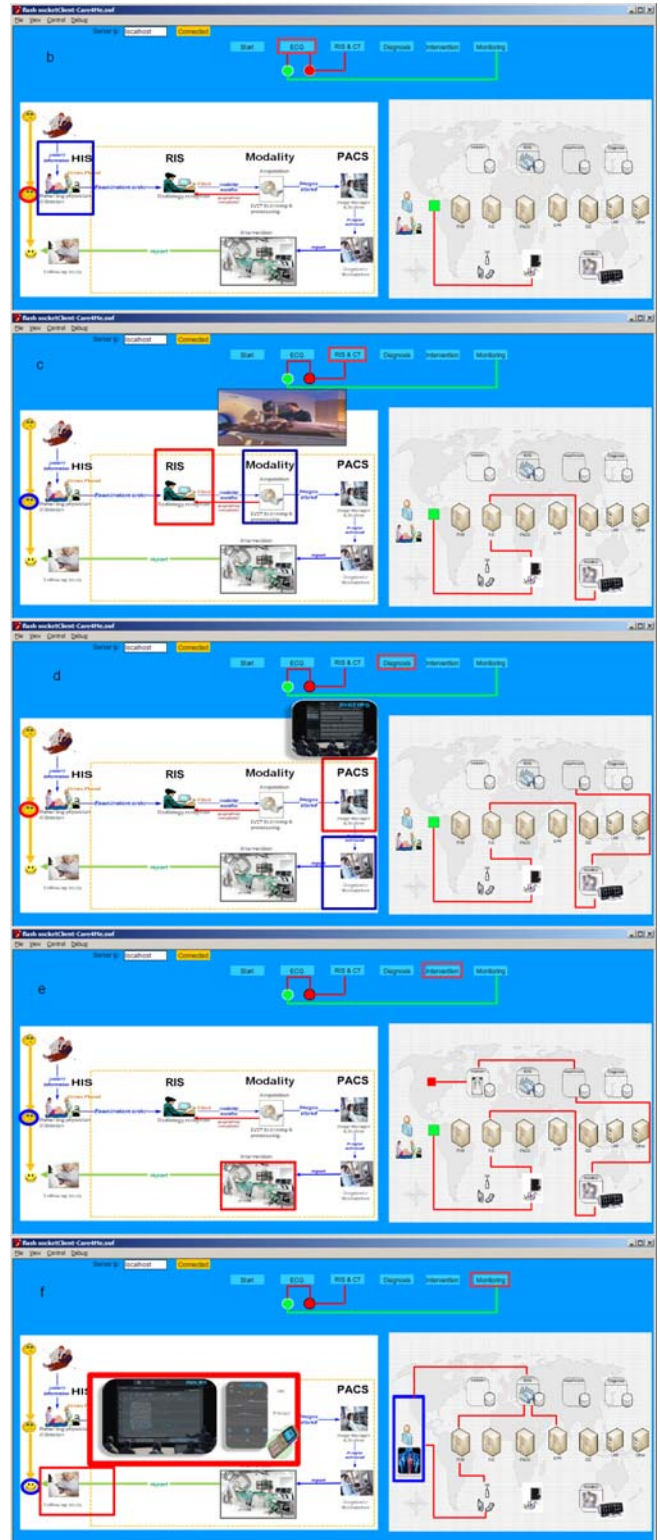
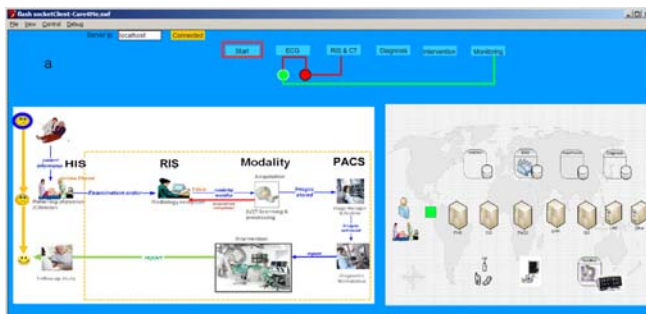


Figure 9. Stakeholder views of Flash animation connected to the POOSL models.

V. CONCLUSION AND FUTURE WORK

As mentioned in the introduction, for complex cyber physical systems it is extremely difficult to ensure that all

parts work together properly. In general, all kinds of specialists and experts in different domains are involved in the development process of such a system. Efficient and effective communication among all stakeholders in the early development phases is crucial to avoid a large number of unexpected problems in the later phases, especially during test and integration.

In this paper a new approach has been developed to bridge the gap between the technical viewpoint on the system architecture and multiple views for non-technical stakeholders. This has been implemented by designing a formal POOSL model of the system architecture, which can be used by technical specialists to validate whether the global system architecture satisfies the main requirements. Moreover, by connecting an animation tool to the executable POOSL model, multiple views on the system behavior are developed for non-technical stakeholders.

Our experience in the healthcare domain shows that this modeling approach quickly reveals all kinds of errors. Our first attempts at constructing an executable architectural model raised many questions about details of the use cases. Later we found many small errors in the architecture definition by reviewing the generated interaction diagrams for all use cases. The Flash animation helps to get feedback from domain experts, such as physicians, about our understanding of the use case and the architectural realization. For instance, after image processing, the resulting images are stored in PACS for the diagnosis and treatment later. However, for real-time image processing it is very important that the intervention can be conducted directly under X-ray guidance with optimal real-time images, instead of storing and retrieving images from PACS. As a result, the POOSL model has been updated according to this feedback.

In future work we intend to extend the POOSL model with timing information for various activities. This enables performance analysis and makes it possible to get insight into the efficiency of the workflows. Other relevant aspects that can be included in the model are privacy and confidentiality of patient data. Moreover, we would like to investigate what types of animation are desirable for different types of stakeholders.

REFERENCES

- [1] OECD Health Data , http://www.oecd.org/topic/0,3373,en_2649_34631_1_1_1_1_37407,0_0.html
- [2] B.D. Theelen, O. Florescu, M.C.W. Geilen, J. Huang, P.H.A. van der Putten and J.P.M. Voeten, "Software/Hardware Engineering with the Parallel Object-Oriented Specification Language, " in Proceedings of the ACM-IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE), ISBN 1-4244-1050-9, IEEE Computer Society, 2007, pp. 139-148
- [3] SHE/POOSL website, see <http://www.es.ele.tue.nl/poosl>
- [4] Adobe Flash Professional, <http://www.adobe.com/products/flash/>
- [5] Socket connections in Flash , programming ActionScript 3.0, http://www.adobe.com/livedocs/flash/9.0/main/wwhelp/wwhimpl/co mmon/html/wwhelp.htm?context=LiveDocs_Parts&file=00000318.htm
- [6] R. Milner, "Communication and Concurrency", Prentice Hall, Englewood Cliffs, New Jersey, 1989.
- [7] Feiler, P.H.; Lewis, B.A.; Vestal, S., "The SAE Architecture Analysis & Design Language (AADL) A Standard for Engineering Performance Critical Systems," Computer-Aided Control Systems Design, 2006 IEEE International Symposium on, 4-6 Oct. 2006, pp.1206-1211
- [8] Feiler P.H., Lewis B., Vestal S., "The SAE Avionics Architecture Description Language (AADL) Standard: A Basis for Model-Based Architecture-Driven Embedded Systems Engineering," In Proceedings of the RTAS 2003 Workshop on Model-Driven Embedded Systems, May, 2003
- [9] Feiler, P.H.; J. Hansson, "Flow Latency Analysis with the Architecture Analysis and Design Language (AADL)," Technical Note CMU/SEI-2007-TN-010, Software Engineering Institute, 2007
- [10] Su-Young Lee; Mallet, F.; de Simone, R., "Dealing with AADL End-to-End Flow Latency with UML MARTE," Engineering of Complex Computer Systems, 2008. ICECCS 2008. 13th IEEE International Conference on, March 31 2008-April 3 2008, pp.228-233
- [11] Systems Modeling Language, <http://www.sysml.org/specs.htm>
- [12] J. Grimson, W. Grimson, and W. Hasselbring, The SI Challenge in Health Care, Communications of the ACM, Vol. 43, No. 6, 2000, pp. 49-55
- [13] K. A. Kuhn, D. A. Giuse, From Hospital Information Systems to Health Information Systems; Problems, Challenges, Perspectives, Method Inform Med, 2001, 40: 275-287
- [14] F. A. Salustri, N. L. Eng, and J. S. Weerasinghe, Visualizing Information in the Early Stages of Engineering Design, Computer-Aided Design & Applications, 2008, 5(1-4)
- [15] P. G. Neumann, Holistic Systems, ACM Software Engineering Notes, 2006