MAFIIA - an Architectural Description Framework: Experience from the Health Care Domain

Erlend Stav, Ståle Walderhaug, Stein Løkke Tomassen,
Lillian Røstad, Nils Brede Moe

SINTEF ICT, 7465 Trondheim, Norway
{ erlend.stav | stale.walderhaug | stein.l.tomassen | lillian.rostad | nils.b.moe }@sintef.no

Healthcare information systems are characterized by having many stakeholders, roles, complex and diverse information systems, high degree of formalized working practices and an intense focus on quality concerns like interoperability, security and reliability. There is an emerging need for a structured architectural tool for supporting system developers and architects working with this kind of critical infrastructure. This paper presents MAFIIA - an architectural description framework specialized for the health care domain. The framework has been used in the development of three very different healthcare information systems: a system for individual care plans, a platform for image-guided surgery and a patient evacuation support system. The experience from the case studies shows that the framework is a useful and flexible tool for creating an architectural description, and assists in keeping the focus on selected quality concerns.

1. Introduction

In health-care organizations a conglomerate of software systems is used. Development and maintenance of such systems are challenging, as special focus is required on security and usability, as well as integration and interoperability with other systems. To handle this challenge, it is essential that the developers and other personnel responsible for the development, maintenance and administration get a good understanding of the system's architecture, its interfaces to environment, and the context in which the system will be used.

For the central concepts of architecture and architectural description we use the following definitions from [1], and for interoperability the definition from [9]:

- Architecture: The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.
- Architectural Description: A collection of products to document an architecture.
- Interoperability: a) The ability of systems, units, or forces to provide services to and accept services from other systems, units or forces and to use the services so exchanged to enable them to operate effectively together. b) The condition achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users.

In this paper we will use information integration as an interoperability mechanism since the same requirements and architectural decisions apply to both.

Unfortunately, architectural descriptions for healthcare information systems (HIS) vary in structure and content – if they exist at all. They seldom include important information like the stakeholders the system was originally built for, which laws and regulations affected the system, which standards that were applied, and which other systems it was built to collaborate with.

From the end users' perspective, successful implementation of a HIS is dependent on the developer's ability to understand the working processes the target system must support. From a high-level viewpoint, a major concern is that the new system must not interfere with other existing systems.

Non-existing architectural descriptions, problems adapting the system to the working processes, and a need for information integration and interoperability was our motivation to develop an architectural description framework called MAFIIA (Model-based Architecture description Framework for Information Integration Abstraction). Collaborative design is encouraged and supported by the framework, to ensure that the systems are built based on real understanding of the needs of the end users and the requirements from environment system interfaces. The framework assures a common structure and content of architectural descriptions for an organization's systems. At the same time, it provides the flexibility to focus on the concerns defined by the organization. This will assist developers in maintenance and evolution as well as development and description of new systems.

This paper presents MAFIIA and our experience from three case studies where the framework was used to develop architectural descriptions of HISs with a special concern for functionality, reliability and interoperability. For each case study we briefly present the background before we summarize our experience from applying MAFIIA to it. In the discussion, we address the following questions based on the case studies:

- How did the prescribed development process of the framework fit the case, and did it help us document what we needed?
- How did the framework address the interoperability concern?
- Was the framework flexible enough to support the individual differences between the cases?
- Was the framework equally suitable when introduced at the start of the development as when used to continue the development of an existing system?

Before describing MAFIIA we first present the background for development of the framework. MAFIIA is presented in section 3 before the case studies are described in sections 4, 5 and 6. The experiences are discussed in section 7 before we provide som concluding remarks in section 8.

2. Background

SINTEF ICT is a research institute in Norway that works with organizations from many sectors, including defense and health. Both the defense and health sectors are characterized by having many stakeholders, roles, complex and diverse information systems, high degree of formalized working practices and an intense focus on quality

concerns like security and reliability. Developing, maintaining and extending such time critical systems, some of which cannot be taken offline, can be very difficult. There is an emerging need for a structured architectural tool for supporting system developers and architects working with this kind of critical infrastructure. An explicit system architecture description framework that includes business aspects as well as traditional systems engineering aspects can help addressing these concerns.

Based on the experiences from architecture description frameworks for command-control systems in the defense sector [2, 3], SINTEF ICT has developed a model based architecture description framework for information systems with a special focus on information integration, called MAFIIA.

The MAFIIA framework adopts the terminology and conceptual framework defined in IEEE 1471 [1], which is a standard of recommended practice for describing the architecture of software-intensive systems. Compared to IEEE 1471, MAFIIA gives further normative guidelines, including the use of UML as notation, a set of predefined viewpoints and a reference architecture.

3. MAFIIA

MAFIIA is a framework for creating architectural descriptions of software intensive systems. The framework assists the architect by:

- Supporting cooperative design through the definition of a set of views and selection of notation that allow end user involvement in important parts of the work.
- Supporting development and description of the architecture of new systems, as well as documentation of the architecture of existing (legacy) systems.
- Providing use guidelines for architectural patterns applicable to systems that need to integrate information from several heterogeneous environment systems
- Providing a structure that ensures that documentation of different systems developed using the framework will have a uniform structure and content.
- Presenting a list of quality related concerns that the architect should consider when creating the architecture, and instructing how to include description of the concerns of particular importance.

An architectural description created using MAFIIA is structured around a set of views, each of which describes the system from a certain viewpoint. Views are useful for illustrating different aspects of the same target system, and are also the basis for RM-ODP [4]. Concerns that are of special importance to the target system, e.g. security and interoperability, must be identified and described. A set of system assets, e.g. standards and laws, that is useful for describing and understanding the architecture is also included. A reference architecture is defined by MAFIIA. This reference architecture can be refined for a specific target system, or for a set of related systems.

It should be emphasized that the main purpose of the architectural description is to give the reader an understanding of the fundamental aspects of the system and its context. Thus, the architectural description is kept at a relatively high abstraction level and does not include e.g. full user requirements, complete business process models, or more detailed design information.

In the following subsections, each part of MAFIIA is described in more detail.

3.1 Concerns

MAFIIA defines how to describe *concerns* of special importance to the system. These concerns will need special attention within all or most of the *views* described in 3.4. A concern may require special models or other formal descriptions to be created to ensure that the architecture description is correct and complete.

Functional aspects that are considered to be of such importance that they should be treated separately and be specifically visible in the documentation should be identified and treated as a concern.

In a HIS, *security* should always be treated as a special concern due to patient privacy issues. Confidentiality, availability and integrity – the key characteristics of information security – are essential in health care information systems. Security should be addressed in a dedicated model in each view of the architectural description.

For a HIS, *interoperability* is a special concern. A HIS must operate in a context where many other critical systems both provide and rely on information from the system being architected. The security concern has a major impact on the interoperability. Single sign-on mechanisms and shared role based access control are requirements that should to be handled by the interoperability concern as well. The focus on interoperability will require the architects to carefully design the information and operation interfaces to the environment, as well as the distribution and realization of the system components.

3.2 System Assets

System assets are sources of information that can be used when developing an architectural description. System assets can be considered as implicit requirements, which are not necessary to include in the requirement view, however assets may be included in component, deployment and realization views. Short descriptions of the most common assets for architectural description of a HIS are:

- **Dictionary**: A dictionary is a reference list of concepts important to a particular model aspect or concern along with discussion and/or definition of their meanings and applications.
- **Standards:** A standard is a formalized model or example developed by a standardization organization or established by general consent. When implementing a HIS, a set of national and international standards will probably be used, and these must be referenced or documented.
- **Laws and regulations:** For a HIS, laws and regulations will affect how the system can be used, and how it has to be built. The architectural description should include references to the laws and regulations that have been considered, including comments on how these apply to the target system.
- **Patterns:** A pattern is a description of a recurring, well-known problem and a suggested solution. Patterns are identified and can be used on many system levels. The MAFIIA framework includes guidelines for when to apply well-known patterns in the architecture. Summary descriptions of recommended patterns are included, along with references to sources such as [5, 6, 7], where the full pattern description can be found. The framework suggests a number of patterns related to interopera-

bility and information integration. The selected patterns are referenced in the architectural description of the target system, and specialized in the view(s) where they are applied.

The use of dictionaries and standards are important for information interoperability between different systems. If two systems should be allowed to interoperate across national or organizational boundaries, they should be in accordance with the same laws and regulations. Correspondingly, the use of architectural patterns will facilitate interoperability between systems and sub-systems.

3.3 Reference Architecture

MAFIIA defines an overall reference architecture for information integration systems. This is a high-level, generic architecture which is used as a basis for development of target system architectures, and to compare architectures of existing systems. The MAFIIA reference architecture defines four logical tiers, and the interface to the environment as shown in Figure 1. The tiers are frequently referred to in the descriptions in the different views.

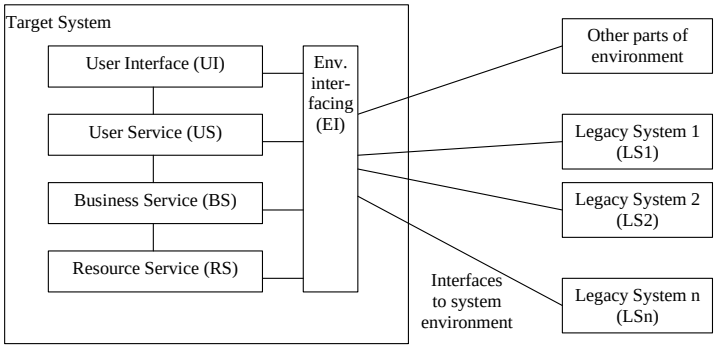

Fig. 1. MAFIIA Reference architecture for information integration systems

3.4 Views and Viewpoints

A central part of MAFIIA is its definition of a set of viewpoints. Each viewpoint defines how a specific view of the target system shall be described, and prescribes a set of models that the view shall include. The notation to use for each model is also defined – normally a set of UML diagrams with accompanying textual description is used. Architectural descriptions created with MAFIIA contain the following views:

- **Context view**: The context view describes the business-related aspects and stakeholders of the target system and its environment. Environment systems that will be involved in or influence the operation of the target system are identified, and their interfaces and collaborations with the target system are described. The context view should be created in collaboration between end users or domain experts, and software architects. The description in this view is important during the initial development of the architecture, but may be even more valuable during maintenance

and integration with other systems, as it provides background motivation for the architecture that may otherwise be forgotten and hard to reconstruct.

- **Requirement view**: The requirement view describes functional and quality requirements that can affect the architecture of the target system. This does not include complete user requirements, but instead generalized versions of each type of user requirement that are of importance to the architecture. The models in this view are based on use case diagrams and tables of prioritized requirements, and are best constructed in collaboration between software architects and end users. Interoperability requirements are derived from the interfacing systems described in the context view, and the framework also provides a set of requirement choices guiding the process of eliciting integration requirements.
- **Component view**: The component view describes the decomposition of the system into components, including their interfaces, interaction, and the information that is handled. The security model is an important part of this view, and describes security mechanisms and how these are integrated with the rest of the system. The models of this view are kept at a logical and platform independent level, and do not include realization details. For this view, the framework presents a set of architectural design issues for information integration systems, and proposes patterns and other solutions that can be suitable when the issue has specific characteristics.
- **Distribution view**: This view describes the logical distribution of components and roles. It describes which components that must be together on a node, which components that must be distributed to different nodes, and which components that may optionally be distributed. The framework includes recommendations for distribution choices based on parameters such as system size, geographical distribution, and communication capacity. The distribution choices can be limited by the current deployment of components in environment systems, as well as their security infrastructure.
- **Realization view**: This view describes how to implement the system described in the other views on a selected target platform. It includes mapping of the architecture to the selected technology platform (e.g. Java or .Net), and also describes the actual deployment of the system on the selected nodes. Both technology platform and deployment choices can be limited by the requirements for integration and interoperability with the environment systems. An important aspect of deploying a new system into an existing information infrastructure includes interoperability testing. The realization view includes a "System Integration Test Model" that describes a set of test scenarios to be conducted during system deployment.

Examples of models from the first four views are included in the case studies in sections 4, 5 and 6.

3.5 Process

The MAFIIA framework recommends an iterative development process. As described in Figure 2, an iteration of the architectural description work usually starts with describing the context view, and ends with the realization view. The work does not proceed in a strict sequence, but frequently returns to previous views when new insight is acquired. Each iteration results in a version of the architectural description that is re-

viewed. More than one iteration may however be necessary to complete the architectural description.

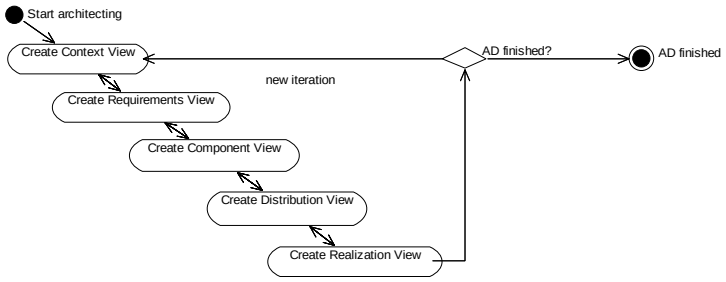

Fig. 2. MAFIIA view description process

4. Case 1: Individual Care Plans

In the SamPro project SINTEF ICT cooperated with the local health care region and the company Visma Unique AS to develop a system for individual care plans. SINTEF ICT was responsible for developing the system architecture, while Visma Unique AS was responsible for implementing the system.

According to the Norwegian health law, patients with long-lasting health problems and with the need for coordinated health services, have the legal right of an individual care plan. The objectives of making an individual care plan are:

- To support a coordinated, unified and individual health service, and ensure a responsible care provider for the patient at all times.
- To map the patient's objectives, resources and needs for different services in different domains and coordinate and evaluate actions to meet those needs.
- To increase cooperation between a service provider, patient and other involved (peer/family), between service providers and other organizations at the same or across levels of service.

A system for individual care plans needs to read, present and report information a variety of health care information systems. To do this it is necessary to identify stakeholders and understand and describe the work processes, user requirements and existing laws and regulations that affect the system, as well as interfaces to the environment systems. MAFIIA was chosen to develop and describe the architecture of the system.

The next section describes how MAFIIA were applied, and results and experience from the use of the method.

4.1 MAFIIA Applied to Individual Care Plans

The users of an individual plan come from different levels of the public health service, and from different health institutions. Some of the users are even outside of the health sector, e.g. patients, relatives and social workers. The main challenge is to en-

sure that the users of the system at all times can access the information they have the right and need to see. The most important concern was security. The most central system assets were relevant laws and standards concerning health informatics and security.

The users participated in developing the Context and Requirement views. Together with the end-users we described the system stakeholders, their relationship to each other and to the new tool. This information was essential for understanding the complex domain, and gave an overview of all the contributors to the requirements. A process model for the new system was developed as part of the context view, to help understand the work-processes a system for individual care plans needs to support. A model describing environment systems was developed to identify candidate systems for integration (see Figure 3). The care plan contains only a short description of the different services a patient receives. The service providers each have a more thorough documentation in their own information systems. The goal was to identify if and how the care plan system should be integrated with these systems. Another integration issue was the use of a common user database. Users, that are health care personnel, should be able to log on to the care plan system using their regular username and password. This requires that the care plan system be integrated with a common user database for health care personnel.

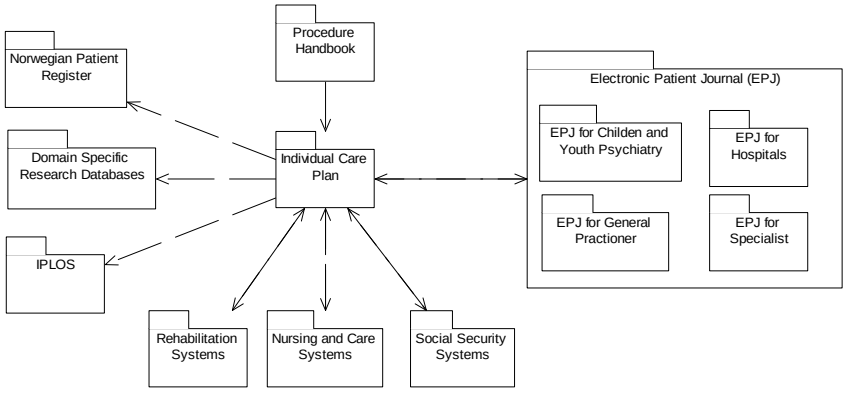

Fig. 3. Individual care plan and environment systems

In component view the system was decomposed into sub-systems and their components. Figure 4 illustrates the main subsystems in an individual care plan system. The access control components were grouped in a separate sub-system that is used by all the other sub-systems. A model for role based access control combined with information categorization was developed as a part of the architectural description.

The system was to be designed as a service in the regional health network. The distribution view described where to place the different logical layers of the system in the network, and the chosen protocols for secure network communication.

.Net was chosen as the implementation platform for the individual care plan system. The realization view therefore included description of mapping to .Net technology components, including mapping of the security mechanisms described in component view to the .Net security model.

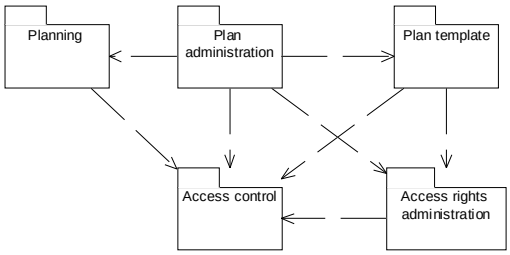

Fig. 4. Care plan system decomposition model

One of the main challenges in the project was to communicate the system architecture to the developers. One person from the development team was involved in architectural meetings to help this process, and after the project ended we had an evaluation meeting with the developers to identify any problems or suggestion for improvement. The meeting confirmed that the use of a structured and model-based framework, such as MAFIIA, resulted in an architectural description that was easy to understand and adopt for the developers, with minimal help from the architectural team.

Visma Unique AS is continuing to work on the system, and is planning to release it as a commercial product on the market in 2005.

5. Case 2: CustusX

CustusX is a platform for image-guided therapy developed by the Center of Competence - 3D Ultrasound in Surgery, which consists of St. Olavs Hospital, Norwegian University of Science and Technology (NTNU), and SINTEF. The goal with CustusX is safer, more accurate, and less invasive surgical procedures.

The initial idea behind CustusX was to have a navigation platform for clinical testing of new procedures in the operating theatre. One of the requirements was that the software should be cross platform executable and run on both ordinary personal computers and in large-scale visualization installations.

Today, the system consists of a dual processor Macintosh computer, a position sensing system, and a navigation software (cross-platform compatible with Macintosh OS X, UNIX, and Microsoft Windows systems). The system is tailored for image-based planning and intra-operative guidance during surgery but can also be used for post-operative control, teleradiology, and simulation or training.

The development of CustusX has been demo driven – that is, repeatedly new functionality has been requested and then implemented. This had led to a system with no structured documentation of the architecture and consequently very difficult to maintain. It was also acknowledged that the system had become too big and complicated, and consequently did not support the needs of the different target groups very well. It was therefore recognized that the system had to be reorganized to make it both more maintainable and configurable to better meet the needs of the different user groups. The first step in this work was to specify an architectural description of the current

system and then for the future system. The resulting architectural description would be of great help for the system architects and the developers both in the further evolvement of CustusX and in making more accurate estimates of the work to be done. MAFIIA was chosen for this task for several reasons; it defines a structure for the documentation, it is user centric, and provides support for HIS development.

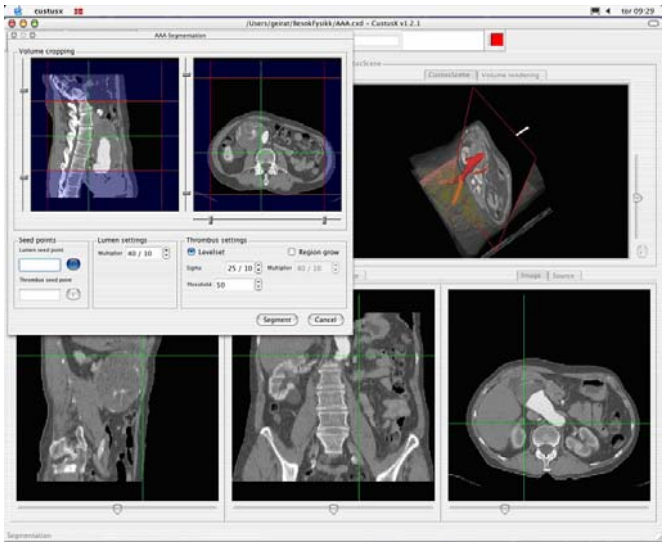

Fig. 5. A screenshot from the CustusX application running on a Macintosh OS X platform.

5.1 MAFIIA Applied to CustusX

CustusX has evolved to be quite an extensive platform for image-guided therapy, which has led to a great number of potential stakeholders that have a strong interest in the use of this system. It was therefore vital to identify all these stakeholders as they may affect the final architectural system description. For instance, some users need the application for telemedicine purposes, e.g. in situations where there is a need of an expert opinion and where the expert is located remotely. This would influence how to describe which components that needs to be distributed.

One important step defined by MAFIIA is to identify all relevant concerns. The concerns identified by the architectural group (including end users) were configurability, performance, reliability, safety, and security. Since CustusX will be used in many different clinical applications like surgery planning, surgery guidance, radiology intervention procedures, radiation therapy, simulation, and training it needs to be highly configurable. The Graphical User Interface (GUI) and the visualization of data should be able to adapt to different users and roles. The system should also be able to integrate seamlessly with environment systems by being able to make use of these systems, e.g. knowing how to communicate with existing Picture Archive and Communication System (PACS). Security was important since the system will contain patient information and be distributed.

Figure 6 shows the distribution view for the CustusX system. The Collaboration management component can be connected to another CustusX system. The PACS server is a legacy system, and is therefore separated in an individual node. Many CustusX systems can access the user database and is consequently located in a separate node.

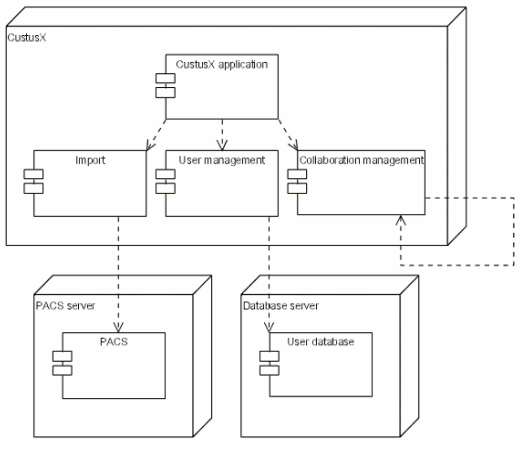

Fig. 6. CustusX system distribution model

The architectural group decided not to specify any realization view in the first version of the CustusX architecture. This view was not needed to make estimates and a detailed plan for the implementation work and will be done in the next version of the architecture.

6. Case 3: Evacuation Support System

In 2002, SINTEF ICT was hired by the Norwegian Ministry of Defense to be the Norwegian project manager in a USA-Norway Memorandum of Understanding (MoU) within the area of telemedicine and new systems and practices of work in the military medical services. Focus was on information flow during evacuation of wounded soldiers.

Today, when injured soldiers are evacuated from the battlefield, medical treatment and observations are documented on a paper-based form that follows the patient. There are several disadvantages with this system:

- The forms tend to become illegible, lost, torn apart, and so on.
- The information only exists at one place, making it difficult for command-control personnel to gain an overview of the injury/casualty situation.
- Each nation in NATO has its own paper record. Information should be interchanged between the countries seamlessly. This is not possible using the paper based form.

A project called Evacuation Support System (EvacSys) was started to test and evaluate a new electronic system for information capture and distribution in a military

medical evacuation. Using personal electronic memory tags, wireless military tactical communication, state-of-the art portable computing terminals and a new military Electronic Health Record (EHR) called SANDOK, a complete distributed information system was specified, developed and tested. EvacSys involved many stakeholders and the system had to be interoperable with both national and other NATO nations' health information systems, as well as the underlying military command and control infrastructure. To handle this concern, MAFIIA was used as a handbook for specifying and developing the EvacSys architecture.

6.1 MAFIIA Applied to EvacSys

The first step in the MAFIIA framework is to identify concerns and system assets. Focus was put functionality, usability, reliability and interoperability. Major system assets were NATO standards and Norwegian laws and regulations for medical documentation and exchange.

The MAFIIA framework's context view was found very useful when describing the context in which the EvacSys will operate. The *Environment Systems Model* and the *Business to System Mapping Model* explicitly illustrates interoperability concerns, thus providing important input to the succeeding viewpoints. Identifying and documenting the environment information and operational interfaces required a lot of time and effort, as much of this documentation was impossible to find.

The Context view models provided an important input to the system requirements models. Figure 7 shows how the different actors interact with EvacSys. Use case and sequence diagrams were used for functional requirements, whereas non-functional requirements had to be expressed in textual rationale descriptions.

As a part of the Components View, the information and processing components were carefully modeled to optimize robustness and integration and interoperability with existing infrastructure. The *System Information Model* was based on national information representation standards, but had to be "abstracted" into a higher level notation to be suitable for use by other systems. The difference between Norwegian and English/American person identifiers and naming conventions required some additional components to be specified.

The EvacSys Distribution View models explicitly express reliability and interoperability related concerns. Critical operational components were distributed onto different logical nodes. The distribution models emphasized that there are no "single point of failure" at the different logical locations (battlefield, mobile aid station, transport etc.) in the EvacSys. For example, all terminals used for information input and output are generic in the way that they have no components that are specialized for one specific user or patient. Information-intensive components such as pulse-oxymetri sensors were put on separate nodes and shared filtered information through standard communication syntax and protocol. The data replication mechanism used to distribute patient data at the field hospitals was implemented according to the push model of the Publisher-Subscriber pattern.

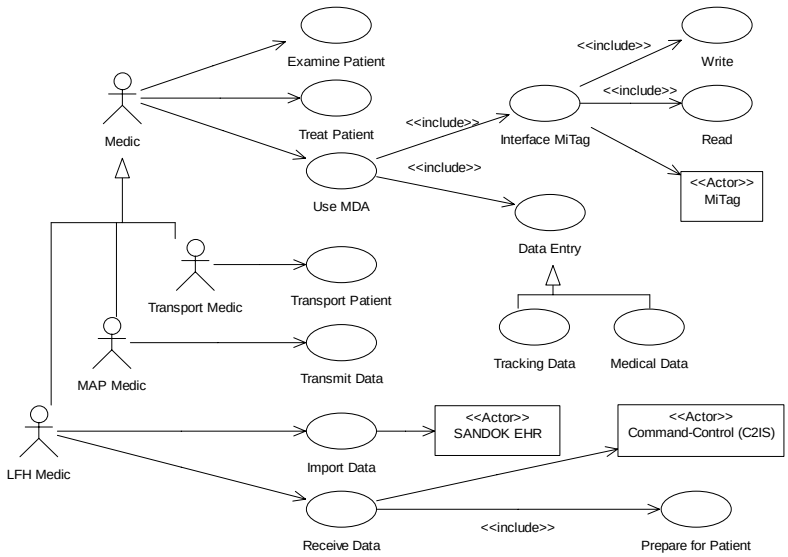

Fig. 7. EvacSys requirements view.

Fig. 8. Military exercise where EvacSys was tested

Military standards and instructions were important input to the realization view description. Only a limited set of equipment, platforms and protocols are approved for use in an operational setting.

The first prototype of the EvacSys system was tested and evaluated during a 5-day military exercise in northern Norway (Setermoen) in December 2003 (Figure 8). The medical battalion of the 6th division conducted several evacuation run-throughs with both the EvacSys system and the traditional paper-based system. The evaluation report concludes that the system architecture and information flow worked according to system requirements.

7. Discussion

This section discusses experiences from using MAFIIA based on the cases presented and addresses the questions stated in the introduction. It should be noted that MAFIIA was used as a tool in the architectural work of the projects, and were not itself the research focus of the projects. The discussion presented here is thus mainly based on a subjective evaluation from the different researchers that participated in the projects, and to some degree on feedback from other participants in the projects. All of the cases are based on projects of approximately 15-25 person months. There is also great variation in type of systems of the cases, even though all of the cases are within the health care domain.

The feedback from the cases indicates that the development process described in MAFIIA was easy and useful to follow. The combination of a description of what to do, and a checklist of what to include of models, standards etc, helped the developers in the work. The assistance provided by the framework in identifying quality related concerns was reported as important to all the cases, e.g. security in all cases and configurability in CustusX. Also the context view was consistently reported to be very valuable in all the cases.

With respect to interoperability concerns, all three cases have been tested in a real environment. Feedback from all of the cases showed that the context view was an essential tool for understanding the complex domains, and gave an overview of all the contributors to the requirements. The EvacSys prototype was successfully tested in an integrated operational setting. SAMPRO was the first system designed for external access within the Norwegian Health Network.

The flexibility of the framework was essential in supporting the necessary variations in the description which the different cases required. The ability to select different concerns to focus on, and the ability to extend each view with new models were utilized in all three cases. There were some differences between the cases in how the requirements were specified. Some used use cases while other preferred only textual descriptions. There were no reports of problems using either.

The framework was found equally useful in the two cases were it was introduced from the start or early in the development process for a new system, and in the case where used for architectural clean-up and further development of an existing system.

8. Related Work

There exist a number of related architectural frameworks that are commonly in use today. RM-ODP (Reference Model of Open Distributed Processing) [4] is a framework that provides the developers a standard for creation of systems that support distributed information processing services to be realized in heterogeneous environments. The method uses five different viewpoints to describe the system. The framework is neutral in the selection of tools for describing the architecture.

TOGAF (The Open Group Architecture Framework) [11] is an enterprise architecture framework that "consists" of a subset of architectures: business, data, application, and technology respectively. TOGAF consists of a set of tools and methods for devel-

oping different architectures. The goal of TOGAF is to become an industry standard method that is neutral to both selection of tools and technologies.

ATAM (The Architecture Tradeoff Analysis Method) [8] is an analysis method used to understand the various tradeoffs that have to be made when creating architecture for software intensive systems.

NATO has started a Multilateral Interoperability Program [10] that focuses on interoperability between member nations' command and control systems.

9. Conclusion

The findings from the case studies indicate that the use of the MAFIIA facilitates development of systems that will operate in a complex environment. Despite the individual differences of the case studies presented here, the framework has proven to provide good assistance for the architectural work, and results in a well-structured architecture description. The method gives excellent support when developing architecture with a strong focus on specific selected concerns, and security in particular.

Applying an architectural description framework like MAFIIA will have best effect if it is used as a standard for all software intensive systems developed within and for the organization. We believe that large organizations, e.g. public health care within a state or country, is in a position where they can require that at least all new system that they acquire or developed are described in a standard of their choosing.

References

1. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE Std 1471-2000. ISBN 0-7381-2518-0.
2. J.Ø. Aagedal, A-J. Berre, B.W. Bjanger, T. Neple, C.B. Roark, “ODP-based Improvements of C4ISR-AF”. 1999 Command and Control Research and Technology Symposium, U.S. Naval War College, Rhode Island, USA. June 29 - July 1, 1999.
3. B. Elvesæter, T. Neple, J.A. Aagedal, R.K. Rolfsen, “MACCIS 2.0 – An Architecture Description Framework for Technical Infostructures and their Enterprise Environment”, Submitted to 2004 Command and Control Research and Technology Symposium.
4. Basic Reference Model of Open Distributed Processing – Part 1: Overview and guide to use the Reference Model. ITU-TS, Rec. X901 (ISO/IEC 10746-1), Standard 1995
5. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns. Elements of Reusable Object-Oriented Software, Addison Wesley, 1995.
6. F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. Pattern-Oriented Software Architecture. A System of Patterns. John Wiley & Sons, Ltd, 1996.
7. D. Schmidt, M. Stal, H. Rohnert, F. Buschmann. Pattern-Oriented Software Architecture. Patterns for Concurrent and Networked Objects. John Wiley & Sons, 2000.
8. P. Clements, R. Kazman, M. Klein, Evaluating Software Architectures: Methods and Case Studies, Addison-Wesley, 2001.
9. Federal Standard 1037C, Department of Defense Dictionary of Military and Associated Terms in support of MIL-STD-188.
10. Multilateral Interoperability Program, http://www.mip-site.org/, accessed 2004-11-30
11. TOGAF, Open Group, web site: http://www.opengroup.org/togaf/, accessed 2004-11-30