Model-Based Design and Implementation of Secure, Interoperable EHR Systems

Bernd Blobel[1], George Stassinopoulos[2], Peter Pharow[1]

[1]University of Magdeburg, Institute of Biometry and Medical Informatics, Magdeburg, Germany, bernd.blobel@mrz.uni-magdeburg.de
[2]National Technical University of Athens, Zografou, Athens, Greece, stassin@cs.ntua.gr

ABSTRACT

For designing and implementing secure, interoperable, portable, and future-proof EHR systems, a comprehensive and standardized methodology supported by appropriate tools has to be established and applied. Based on the component paradigm, the ISO Reference Model – Open Distributed Processing has been used to describe the different views on information systems deploying the appropriate vocabulary for each single model view. The concepts considered rank from legal, organizational, and functional up to technical aspects of systems. The harmonization of vocabularies can be performed by meta-languages. The approach has been demonstrated for the model-based design, implementation and maintenance of a clinical study distributed over the Internet.

INTRODUCTION

For establishing efficient and high quality care for patients, health networks with an electronic health record (EHR) as core application must be designed for enabling trustworthy interoperability between different healthcare establishments (HCE) directly and indirectly involved in patient's care. This interoperability has to be provided at knowledge level meeting legal, ethical, and organizational requirements in a flexible and portable way including multimedia and mobile devices.

In that context, methods and tools have to be established to enable formalization and structuring of components needed as well as to realize their management.

ISO REFERENCE MODEL – OPEN DISTRIBUTED PROCESSING

Information systems have been designed, developed and implemented for the purpose of supporting special business objectives and goals. Therefore, that information must be defined which is needed in the business model's context. This information has to be aggregated and processed in a proper way and must be implemented at a specific platform. To compare the architectural, functional, methodological, and technological framework of information systems, the ISO Reference Model – Open Distributed Processing (RM-ODP) can be used[1]. This reference model defines possible views on systems such as Enterprise View, Information View, Computational View, Engineering View, and Technology View.

THE GENERIC COMPONENT PARADIGM FOR FUTURE-PROOF HEALTH INFORMATION SYSTEMS ARCHITECTURE

Regarding future-proof health information systems and health networks in general, we have to look for concepts in structure and function expressing domain knowledge, but also for concepts of security, safety, and quality. Additionally, systems and their components have to be considered within their contextual frameworks regarding legal, organizational, functional, and other aspects. Considering security issues, the concepts of communication security can be distinguished from application security. Quality and safety are related to the latter one. Within a concept, different levels of granularity and abstraction can be defined forming a layered model of services, mechanisms, algorithms, and data[2].

According to the generic component model[3], all views, information content, functionality, implementation environment, and underlying technology but also the proper level of granularity might be modeled in a consistent way. In this way the services and the complexity of the running application component can be defined according to the application environment and the user needs. Services concern entry, processing, and presentation of data but also the enforcement of underlying policy for communication and cooperation. The generic component model enables claims change management (viewpoint of the system) and the resolution of the component's complexity by the transition to less complex sub-components as shown in figure 1. Each specific model in the abstraction-granularity space reflects one specific archetype. A theoretical consideration on consistency of state transitions within the generic model has been provided[3].

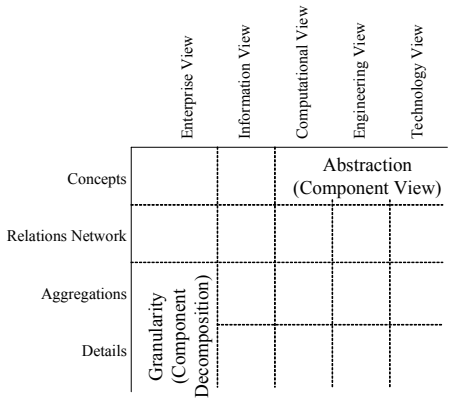
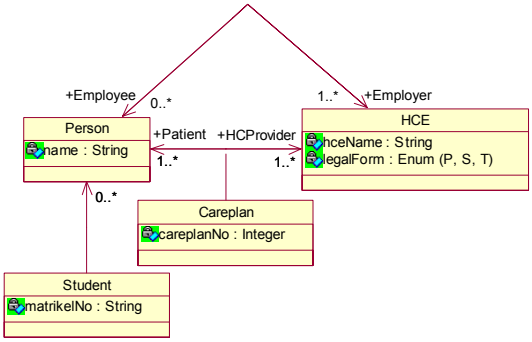


Figure 1. State Transitions within the Abstraction-Granularity Matrix of Component Systems

The description of the components is established in UML models constraint on the different views of the RM – ODP[1]. According to the business needs expressed in business and workflow models and sub-models, the information required including the computational aspects of its package structure is modeled using class diagrams, sequence diagrams, activity diagrams, package diagrams etc. Describing the engineering aspects of platform-specific models, the protocols finally represent the technology aspects. For running systems, all views in their relation have to be considered which requires the harmonization of the different vocabularies used for modeling. This has been performed using meta-languages defined by the XML (Extensible Markup Language) standard set[4]. Related to granularity and technology viewpoint, mobile computing has to meet special requirements which are easily enabled by this dynamic selective approach of the proper state of a complex system. In that context, negotiation and enforcement between concepts have to be performed, consistency and process relationship (the latter supported by the ISO General Relationship Model[5]) have to be ensured. Algorithm and tools to enable such services have been developed and will be further improved.

To transform graphical vocabularies into XML specifications, XML Metadata Interchange (XMI) has been applied[6]. Figure 2 shows an example for transferring an UML class diagram into an XML Data Type Definition (DTD)[4]. Meanwhile, XML Schemata increasingly replace DTD[7]. The approach corresponds to OMG's Model Driven Architecture (MDA)[8]. Beside UML and XML tools for specifying the platform-independent models, special tools have been developed within the HARP project to specify platform-specific models and to implement them in a Java-based Web application.




```
<!ELEMENT Exp ((Person | Student | HCE | CarePlan)*) >
<!ELEMENT Person.name (#PCDATA)>
<!ELEMENT Person.HCProvider (HCE)* >
<!ELEMENT Person.Employer (HCE)* >
<!ELEMENT Person (Person.name?, Person.HCProvider*,
        Person.Employer*)? >
<!ELEMENT HCE.hcename (#PCDATA) >
<!ELEMENT HCE.legalForm EMPTY>
<!ATTLIST HCE.legalForm
        xmi.value (P | S | T)  #REQUIRED>
<!ELEMENT HCE.Patient (Person | Student)* >
<!ELEMENT HCE.Employee (Person | Student)* >
<!ELEMENT HCE (HCE.hcename?, HCE.legalForm?,
        HCE.Patient*, HCE.Employee*)? >
```

Figure 2. Transferring UML into DTD

THE HARP ARCHITECTURAL APPROACH

The HARP[9] (Harmonisation for the Security of Web Technologies and Applications) project (Project Number: IST-1999-10923) was funded by the European Commission as part of the 5th Framework Programme's Information Society Technologies (IST). Within this project, partners from Greece, Germany, Norway, United Kingdom, and the Netherlands have specified, developed and implemented the HARP Cross Security Platform (HCSP) for Internet based secure component systems. The development methodology as well as development tools needed have been specified as described in the following section.

1 The HARP Cross Security Platform. The HARP project's objective is building up entirely secure applications in client server environments over the Web. Real interoperability leads to a closer connection of both communication and application security services. Communication security services comprise strong mutual authentication and accountability of principals involved, integrity, confidentiality and availability of communicated information as well as some notary's services. As a result of the authentication procedure, authorization for having access to the other principal has to be decided. Application security services concern accountability, authorization and access control regarding data and functions, integrity, availability, confidentiality of information

recorded, processed and stored as well as some notary's services and audit.

To provide platform independence of solutions in HARP as a real three tiers architecture, the design pattern approach of developing a middleware-like common cross platform (HCSP) has been used. In HCSP, platform-specific security features have been isolated. Using an abstraction layer, communication in different environments is enabled. According to the component paradigm, an interface definition of a component providing a platform-specific service specifies how a client accesses a service regardless how that service is implemented. So, the HCSP design isolates and encapsulates the implementation of platform-specific services behind a platform-neutral interface. It also reduces the visible complexity. Only a small portion has to be rewritten for each platform. The solutions concern secure authentication as well as authorization of principals even not registered before deploying proper Enhanced Trusted Third Party (ETTP) services[9]. Especially, it helps to endorse policies by mapping them on processing components. For that reason, HARP components follow the specification of generic components[3].

HARP's generic approach implements several basic principles. HARP's solution of embedding security into virtually any application to be instantiated over the web-based environment outlined above is based on object oriented programming principles. It is based on Internet technology and protocols solely. The trustworthiness needed has been provided by applying only certified components which are tailored according to the principal's role. In fine-grained steps, the solution establishes its complete environment required, avoiding any external services possibly compromised. After strong mutual authentication based on smartcards[10] and TTP services[11], the security infrastructure components are downloaded and installed to be used for implementing the components needed to run the application as well as to transfer data input and output. The SSL (Secure Socket Layer) protocol deployed to initiate secure sessions is provided by the Java Secure Socket Extension API. The applets and servlets for establishing the local clients and the open remote database access facilities communicate using the XML standard set including XML Digital Signature[4]. Because messages and not single items are signed, the messages are archived separately for accountability reasons meeting the legislation and regulations for health.

Policies are dynamically interpreted and adhered to the components. All components applied at both server and client site are checked twice against the user's role and the appropriate policy: first in context of their selection and provision and second in context of their use and functionality.

Applet security from the execution point of view is provided through the secure downloading of policy files, which determine all access rights in the client terminal.

This has to be seen on top of the very desirable feature that the local, powerful, and versatile code is strictly transient and subject to predefined and securely controlled download procedures. All rights corresponding to predefined roles are subject to personal card identification with remote mapping of identity to roles and thereby to corresponding security policies with specific access rights.

For realizing the services and procedures described, an applet consists of the sub-components GUI and interface controller, smart card controller, XML signing and XML processing components, communication component applying the Java SSL extension, and, last but not least, the data processing and activity controller. Beside equivalent sub-components and an attribute certificate repository at the server side, policy repository, policy solver, and authorization manager have been specified and implemented as a "light weight" Resource Access Decision service (CORBA: RAD)[8].

After exchanging certificates and establishing the authenticated secure session, servlet security is provided from the execution point of view through listing, selecting and finally executing the components to serve the user properly. By establishing an authenticated session that persists for all service selections, a single-sign-on approach can be realized.

A policy tool establishes the user interface (applet) with the user-related authorization. The HARP administration tool facilitates the definition of information and function of the servlets applied.

HARP enables the implementation of openEHR in a convincing way. Using the open environment of certified Java™ components, portability of the HARP solution to any platform is guaranteed.

In the server-centric approach, a web-accessible middleware has been chosen based on its support of basic security functionality, e.g., MICO/SSL, Apache Web server with mod_ssl, Apache JServ, and Apache Jakata Tomcat.

Combining the server-centric approach of HCSP, its server-centric approach and the network-centric VPN behavior, the completely distributed HARP Cross Security Platform has been designed.

2 Completion of Platform-Specific Modeling and Integration through XML. As partially mentioned already, a set of XML-based tools has been developed for platform-specific modeling purposes. Using variants of the same generic Design XML, role-related and rule-related GUIs are instantiated according to implemented sophisticated access policies, both using View XML in a dynamic way. For passing data to and from the instantiated GUI, Data XML has been specified. The access to legacy or new data-

bases has been defined by Associations XML. The Design XML will be completed by the XML-based instantiation of functional components.

The instantiation of a business application practically involves the execution of three steps:

In the context of a specific business application, all the information available to the end user is defined in a structured way. Defining the presentation of information to the end-user by using the *Policy Tool*, the administrator constructs trees that contain information categories and information fields and define their respective properties.

For retrieving information from databases relative to the information structure defined in the previous step, the *Administration tool* enables the definition of associations between information fields and database fields as well as of database navigation rules in a visual way.

Using the *Policy Tool,* the aforementioned information structure is constrained by authorization and visualization information according to the user profiles.

3 Data Structures. Integration is based on common data structures controlling the instantiation and the overall functionality of the platform. Each business application developed in the HARP platform context is based on a starting XML template, which describes all kinds of information possibly presented and manipulated by all potential application users according to their roles. Figure 3 exemplifies the information structure DTD.

```
<!ELEMENT informationstructure
(informationcategory +)>
<!ELEMENT informationcategory
(informationcategory +|informationfield+)>
<!ELEMENT informationfield EMPTY>
<!ATTLIST informationcategory
   name CDATA #REQUIRED
   abstract (true|false) #REQUIRED>
<!ATTLIST informationfield
   name CDATA #REQUIRED>
```

Figure 3. Information Structure DTD[12]

The information structure model is based on the concept of information fields belonging to information categories. For each business application an instantiation of the above structure has to be created. This data structure serves as a general template for the generation of all other data structures, which are involved in the specific application.

By additional information about authorization and visualization, the information structure is constraint to the profile structure. A corresponding DTD example is given in figure 4.

```
<!ELEMENT informationstructure
(informationcategory +)>
<!ELEMENT informationcategory
(informationcategory +|informationfield+)>
<!ELEMENT informationfield EMPTY>
<!ATTLIST informationcategory
   name CDATA #REQUIRED
   visualcomponent CDATA #REQUIRED
   abstract (true|false) #REQUIRED>
<!ATTLIST informationfield
   name CDATA #REQUIRED
   visualcomponent CDATA #REQUIRED
   haspredefinedvalues (True|False) #REQUIRED
   editable (True|False) #REQUIRED>
```

Figure 4. Profile Structure DTD[12]

Another information structure constraint concerns the data structure of the actual information exchanged between client and the server during a session.

The association information structure entails the adaptability to use the platform with a large family of legacy databases. It contains information concerning the association of the data transferred between the client and the server (profile structure fields) as well as the database fields of the application data sources. For space saving, the DTD examples for the latter two data structures which are based on the same approach and therefore easy to design haven't been presented here.

The aforementioned specifications describe the structure of HARP components to be defined for instantiation. For completing the specifications, behavioural aspects have to be added concerning component relationships, time constraints, etc., by using both UML behaviour diagrams and UML extensions.

Figure 5 presents the HARP implementation environment.

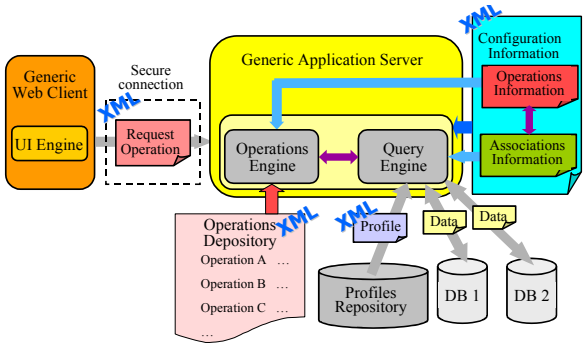


Figure 5. The HARP Implementation Environment[12]

4 Harmonizing the HARP Approach and OpenEHR. The HARP approach enables the implementation of any EHR component following constraints defined in archetype models and expressed in XML schema[13,14]. The HCSP facilitate the instantiation of those components by combining both specifi-

cations in the sense of certified components. So, any granularity, any constraint in the sense of domain knowledge, organizational structure, underlying policy, technological requirements for structure and presentation providing portability, etc. are supported properly.

The development of HARP rules underlying the XML messages which themselves establish the HARP components (servlets and applets) is based on UML models. Within the HARP project developed independent from openEHR, these models reflect archetypes[13,14,15]. For enhancing the current openEHR specification by security archetypes, a harmonization in concepts and especially terminology used needs to be performed.

The momentary weakness of the HARP approach has been the immature tooling for developing the platform-independent models, integrating the different views as well as the platform-specific transfer of those models. The transfer from one model-specific vocabulary to the other has been done manually using the human intelligence instead of formal methods. Beside the enhancement of the modeling methodology demonstrated by concrete components, the tooling for harmonizing the model views will be provided within a successor project. For that purpose, EHR constraint models expressing the information view of concepts and their computational aspects, EHR-related business models, but also engineering views deal with the EHR environment (database, operation system, etc.)

CONCLUSIONS

EHR architecture[16] and subsequently specified and implemented EHR systems have to meet the shared care paradigm establishing openness, interoperability, scalability, and portability for providing any needed and permitted information to any authorized user at time, location, and format required, including mobile devices. Furthermore, EHR systems have to comply with comprehensive security solutions solely based on available and emerging standards. Actual EHR architecture standards comparably presented in the paper move in the direction requested. Emerging common projects harmonize the different approaches towards a "global" openEHR[14,15,16].

The European HARP project has specified and implemented open portable EHR systems enriched with enhanced TTP services and comprehensive development strategies for establishing fine grained application security services. Constraints specified can be bound to components at runtime, enabling different views or supporting specific domain knowledge concepts. By binding attribute certificates to components, appropriate policies can be enforced. These constraints such as, e.g., certificates are interpreted at both server and client side using authorization services. The HARP Cross Security Platform is solely based on standards including the XML standard set for the establishment of EHR clients and servers as well as their communication.

REFERENCES

1. ISO/IEC 10746-2 "Information technology – Open Distributed Processing – Reference Model: Part 2: Foundations".
2. Blobel B, Roger-France F. A systematic approach for analysis and design of secure health information systems. *International Journal of Medical Informatics* 62 (3) (2001) 51-78
3. Blobel B. Application of the component paradigm for analysis and design of advanced health system architectures. *International Journal of Medical Informatics* 60 (3) (2000) 281-301
4. World Wide Web Consortium. XML specifications. http://www.w3c.org
5. ISO/IEC 10165-7 "Information technology – Open Systems Interconnection – Structure of management information systems: General relationship model"
6. Jeckle M. Entwurf von XML Sprachen. *Java Spectrum* 6/2000, 56-60
7. Health Level Seven, Inc.: www.hl7.org
8. Object Management Group, Inc.: CORBA Specifications. http://www.omg.org
9. HARP Consortium: http://www.ist-harp.org
10. CEN TC 251 ENV 13729 "Health informatics - Secure user identification – Strong authentication using microprocessor cards (SEC-ID/CARDS)", 1999
11. ISO DTS 17090 "Public Key Infrastructure, Part 1 – 3", 2001.
12. Vlachos M, Stassinopoulos G. Open, flexible and portable secure Web based health applications. In Blobel B, Pharow P. (Edrs.) *Advanced Health Telematics and Telemedicine*, pp 41-50. Volume 96 Studies in Health Technology and Informatics. Amsterdam: IOS Press, 2003
13. Beale T. An interoperable knowledge methodology for future-proof information systems, 2001
14. openEHR Foundation. http://www.openehr.org
15. GEHR Project: www.gehr.org
16. CEN ENV 13606 "Health informatics – Electronic healthcare record communication", 1999

Acknowledgements

The authors are in debt to the European Commission for funding as well as to the EHR community and to the HARP project partners for support and kind cooperation.