A Dynamic Product Line for an Electronic Health Record Management System for Cancer Care [1]

Matthias Coym, Bernhard G. Humm, Patrick Spitzer
Department of Computer Science
Hochschule Darmstadt – University of Applied Sciences
Haardtring 100, 64295 Darmstadt, Germany
bernhard.humm@h-da.de

Paul Walsh
NSilico Lifescience Ltd.
Bishopstown, Co. Cork, Ireland
paul.walsh@nsilico.com

Keywords: Product Line Engineering, Medical Information System, Electronic Health Record, HL7

1. Introduction

The field of medical information technology is advancing, standards are maturing, and systems such as electronic health records (EHRs) are increasingly being adopted (Yamamoto et al., 2012). However, clinical IT requirements are also continually evolving and expanding rapidly (Anhøj, 2003; Bates et al., 2014). Therefore, more efficient approaches are needed to deal with large amounts of EHR data while flexibility is needed to cope with the rapid evolution and expansion of diverse data models from different disease domains which is in turn driven by evolution in clinical care. Moreover the dynamic nature of medical informatics requirements can cause a prohibitive cost when it comes to integrating new features, such as customising treatment pathways, supporting multi-tenancy deployments and integrating and interoperating securely with third party eHealth systems. Each new eHealth software feature may requires both the extension of the data model and the potential need to integrate data with external IT systems, while maintaining trust and patient privacy, so as to provide a similar set of features across a range of deployments.

Therefore a medical information system for managing *electronic health records (EHR)* must be highly dynamic due to ever changing medical methods and practices. One way of dealing with dynamic requirements is to use a software product line engineering approach, which entails the use of software engineering methods, tools and techniques for creating a suite of software that share a common, managed set of features, from a shared set of software assets using a common means of production, to efficiently meet requirements of a particular sector[2].

The concept has emerged from the automotive industry where a product line of different cars models can be efficiently manufactured, from a common set of component parts. Application of this *product line* approach for EHR management must be even more dynamic in order to configure different clinical practices in various hospitals. This paper presents a dynamic

[1] This work was funded by the European Commission, Horizon 2020 Marie Skłodowska-Curie Research and Innovation Staff Exchange, under grant no 644186 as part of the project SAGE-CARE (SemAntically integrating Genomics with Electronic health records for Cancer CARE).

[2] Software Product Lines Carnegie Mellon Software Engineering Institute Web Site

product line for an EHR Management System for Cancer Care. EHR evolution is facilitated by the HL7 Reference Information Model[3]. A sophisticated rights management is being used for configuring tenant-specific services.

2. Problem Statement

The described concept shall outline the architecture of a product line for an EHR Management System in cancer care. The product line is not designed for the mass market.

The following fundamental requirements were identified:

1. Dynamic derivation of concrete products:
   Concrete products shall be derivable from a software platform via configuration, covering all possible variants. Meaning that various medical specialties, medical information services and technical services have to be considered. Furthermore, configuration may concern every layer of the architecture.

2. Configurability on different levels:
   Products meet the requirements of a customer on different levels, e.g. a default configuration for a hospital and a differing specific configuration for a single physician. Consequently, where appropriate, different configuration levels must be considered.

3. Multiple products on one instance:
   The product line must be able to derive multiple products, each with its own configuration and data, on one running product line instance. It is mandatory, that concrete products must not know if they share their product line instance with other products.

4. Fundamental security mechanisms:
   Only valid users shall have access to their application. They should have their own account with its own corresponding roles and rights. For example a melanoma physician must not be able to view patients, which are not assigned to his area of responsibility.

5. Change Tracking:
   Changes to the persistent data in the database shall be traceable. Additionally, data must not be deletable to ensure compliance standards.

6. Flexible data model:
   To meet the needs of all products, the data model must be flexible and extensible. Negative effects of the flexibility on efficiency and convenience shall be counteracted.

7. Flexible user interfaces:

[3] http://www.hl7.org/implement/standards/rim.cfm

A flexible user interface for EHR management with reusable elements shall be designed, enabling to build up patient data dynamically.

3. Concepts

3.1. Architecture

The product line architecture is structured in three layers, which are loosely coupled. An overview of the architecture is shown in Figure 1.

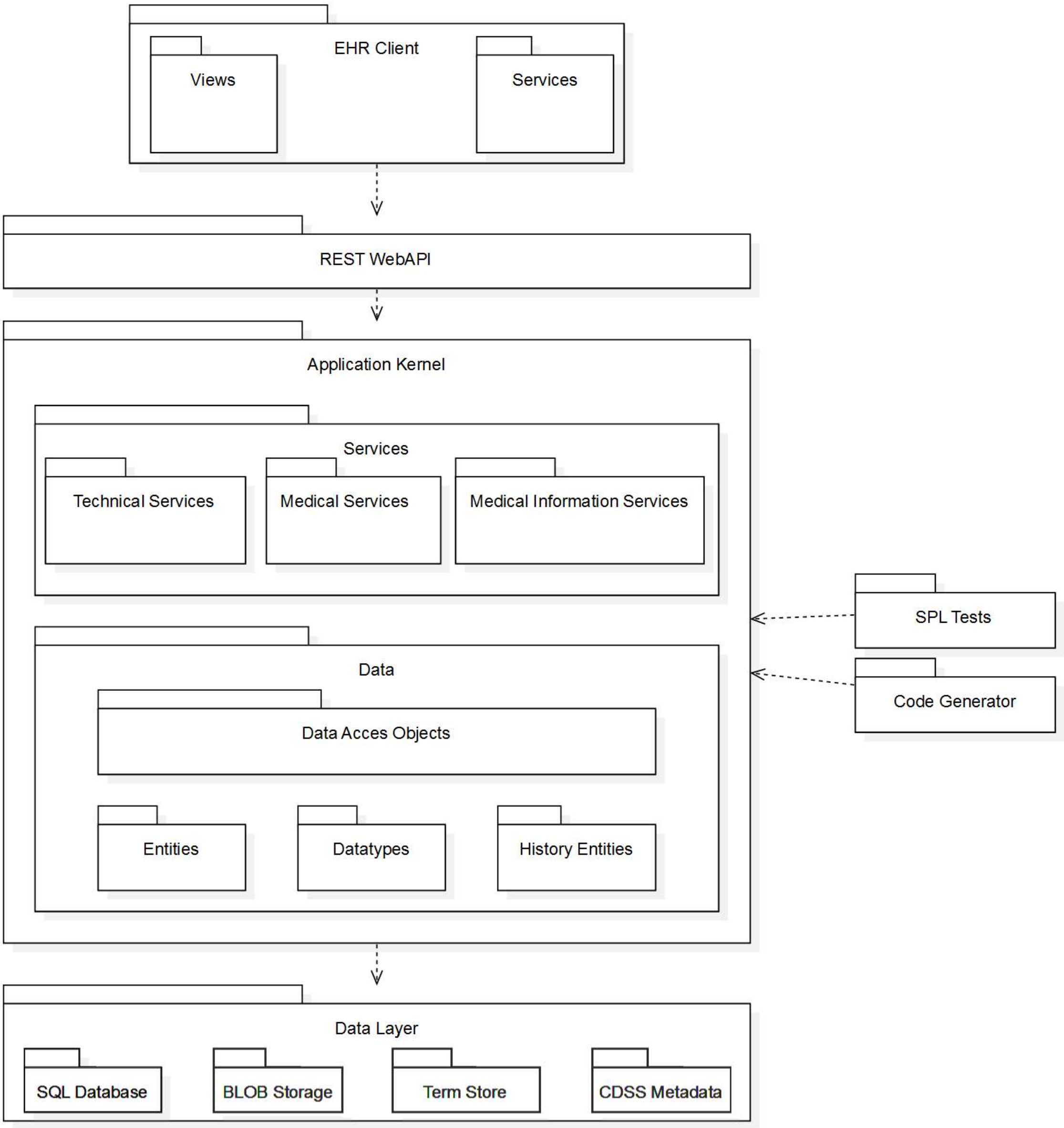

Figure 1: Product Line Architecture Overview

The top layer is the *EHR client*, which is the configurable client application for all products of the product line. It is designed as a web client which is loosely coupled via a *REST WebAPI*. The view package contains view components, being made up of the graphical user interface

(GUI) and corresponding presentation logic. The service package contains reusable client logic, shared by multiple view components. The next layer consists of the *REST WebAPI* and the *Application Kernel*. The *REST WebAPI* provides an interface for the client and is designed according to the Representational State Transfer (REST) principle (Fielding, 2000). The *Application Kernel* contains the business logic of the application. It is made up of different types of services and a data access layer. The data access layer uses data access objects (DAOs) and an object-relational mapper (ORM) for accessing a relational database. The packages *Entities*, *DataTypes* and *HistoryEntities* contain the data items. The bottom layer contains the persistent data of the Software Product Line. It is made up of different storage types. The relational *SQL Database* stores the EHR data, the *BLOB Storage* is used for storing files, e.g. images, while the *Term Store* and *CDSS Metadata* are used for the *Medical Information Services*. See (Beez et al., 2015) and (Idelhauser et al., 2016) for further details.

3.2. Configuration Hierarchy

The product line can be configured on three different levels, as shown in Figure 2.

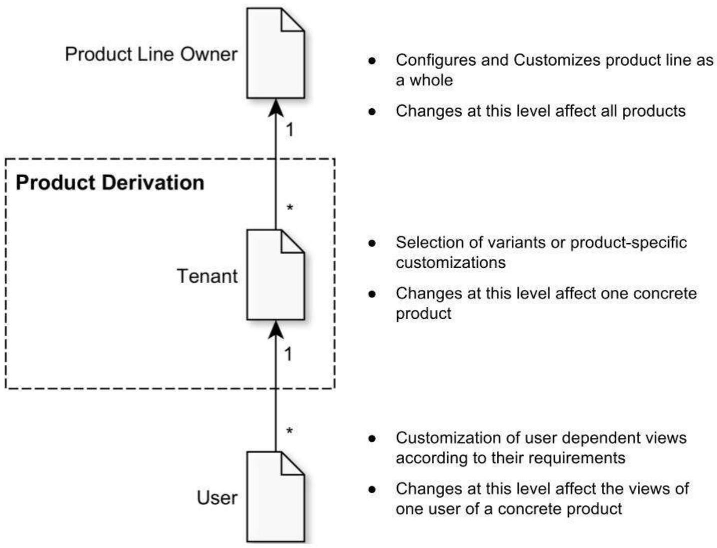

Figure 2: Configuration Hierarchy

At the highest level the *Product Line Owner* decides which features are supported by the Product Line in the first place. At the second level the product derivation takes place. The *Tenant* selects specific features from the product line called variants to create a specific product according to its needs. A tenant may be a hospital group, an individual hospital, or a hospital department. At the lowest level, each user of the product has the option to customize the views according to his or her needs.

In a classic product line, the product derivation takes place at compilation time. In a dynamic product line this must be possible at runtime. To make this possible, the configuration is tightly integrated in the access management. Tenant rights are given according to the chosen variants. The individual user rights are a subset of the tenant rights. When a user logs in with his or her credentials, the configuration is loaded and the correct views are shown. If a user has access to more than one medical service, he is able to switch between these services.

3.3. Access Management

For the access management of the dynamic product line, we use the approach of *role-based access control* (Ferraiolo and Kuhn, 1992). The implementation is additionally based on the security, authentication, and authorization documentation for ASP.NET Web API (Anderson, Pasic and Dykstra, 2012).

The attributes of a user are bundled in an object called the user account, which contains a userID as the unique identifier. For the authentication process, a user has to provide his or her userID and a credential to the system. The credential provides evidence for the user's claimed identity, e.g. a password which is only known to the user and the system. If userID and credential match, the user is authenticated. A security context or token can be bound to the user's connection as evidence that the user has already been authenticated. After the authentication, the requests of the user can be authorized. Authorization can be defined per operation or resource that an interface offers and is realized via a role based access control model.

Access permission to the EHR system depend upon two factors. The first are the permissions granted to individual users are, as usually, according to their role in the hospital as shown in Figure 3. An administrator for example might have the right to manage users but should not be able to access the patient data.

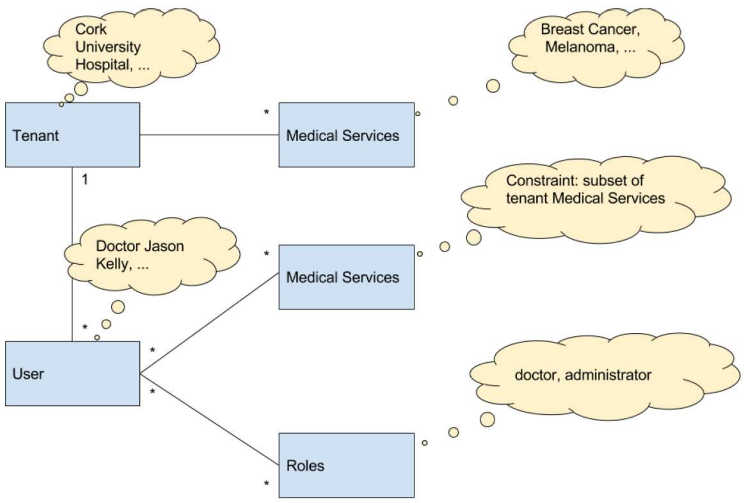

Figure 3: Access Management

Additionally, there are permissions granted to individual users according to their tenant's access permissions like the access to medical services. As described in Section 3.1, a tenant's product is derived from his configuration of variability points. For example, the Cork University Hospital might have a product with the breast cancer medical service, but not the melanoma medical service. Therefore a consultant from Cork University Hospital is not able to access the melanoma medical service.

Apart from the restrictions at the application layer and client layer, access is also restricted at the database. It must be ensured that no tenant can access data of another tenant. To ensure this, each database entry is enhanced by a tenantID. Every database access is performed with this tenantID as an extra restriction. This approach is called *row level security* or *shared table* (Jacobs and Aulbach, 2007).

3.4. Flexible Data Model

The flexible data model is based on HL7 Reference Information Model (RIM) (ISO, 2006). This offers the following advantages:

1. It is a common, maintained, and well documented ISO-Standard for health care information systems.
2. The model adds the flexibility to extend the data model without changing the relations between entity classes.
3. The HL7 RIM appends the ability to create new links between concrete objects at run-time of the application through the usage of association classes.
4. Compatibility to other systems is ensured, which implement the HL7 RIM and therefore between all products of the software product line.

See Figure 4.

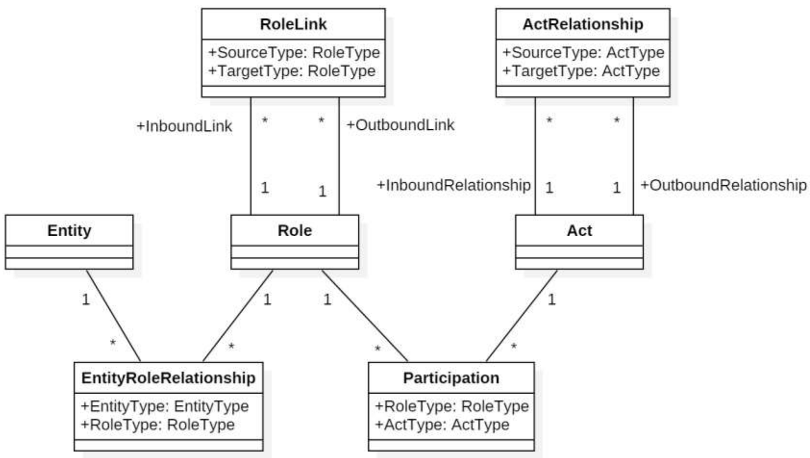

Figure 4: Flexible Data Model oriented at HL7 RIM

In HL7 RIM according to (ISO, 2006), the association between entity and role is not realized with an association class. In order to be consistent with the other association classes *Participation*, *RoleLink* and *ActRelationship*, we introduced a new association class called *EntityRoleRelationship*. This allows dynamically changing relationships between entities, roles and acts, while still being compatible with HL7 RIM.

With the flexible data model it is possible to support a rich variant of products, while still allowing tenant-specific associations. For example, one hospital might prefer to associate a medication directly with the patient while another hospital prefers, instead, to associate it with the cancer issue attached to the patient. Both preferences can be realized without modifying data model or any database table.

Also refer to Humm and Walsh (2015).

4. Related Work

Neither the combination of *Dynamic Software Product Line* with *reconfigurable multi tenant aware Software as a Service* applications, nor *EHR Management System*s (EHRMS) are new research fields. The combination of an *EHR Management System* with a *Dynamic Software Product Line*, however, is still an unexplored field.

Bahga and Madisetti outlined a cloud-based approach for the design of an interoperable EHRMS (Bahga and Madisetti, 2013). They describe the architecture for an EHRMS, but with the explicit focus on semantic interoperability, data integration, and security. Albeit, the paper focuses on single-system engineering instead of creating a product line.

Kuo (2011) summarizes the general opportunities and challenges of cloud computing to improve health care services, to benefit healthcare research and to change the face of health information technology.

The SPLiCE (Software Product Line for healthCarE) project proposes a model-driven engineering method for healthcare information systems. A Software Product Line shall thereby be created, which integrates clinical data models, described according to the "openEHR" specifications, and architecture models, specified in the "Acme" architecture description language (Gomes et al., 2012).

5. Conclusion and Future Work

In this paper, we have presented a dynamic product line for an EHR Management System for Cancer Care. EHR evolution is facilitated by the HL7 RIM. A sophisticated rights management is being used for configuring tenant-specific services.

We have implemented the product line concept prototypically and applied it to three medical services: melanoma treatment, breast cancer treatment, and a receptor database. The initial results are most promising and the reception by clinical consultants is positive. Therefore, it is planned to further develop the prototype to a productive system.

Towards this end, future work is needed. After completing development, extensive testing is required including performance testing. Furthermore, we plan an extensive usability study with end users.

References

Anderson, R., Pasic, A. and Dykstra, T. (2012). *Security, Authentication, and Authorization in ASP.NET Web API*. [online] Docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/aspnet/web-api/overview/security/ [Accessed 1 Sep. 2017].

Anhøj, J. (2003). "Generic Design of Web-Based Clinical Databases". In: *Journal of Medical Internet Research*, 5(4), p.e27.

Bahga, A. and Madisetti, V. (2013). "A Cloud-based Approach for Interoperable Electronic Health Records (EHRs)". In: *IEEE Journal of Biomedical and Health Informatics*, 17(5), pp.894-906.

Bates, D., Saria, S., Ohno-Machado, L., Shah, A. and Escobar, G. (2014). "Big Data In Health Care: Using Analytics To Identify And Manage High-Risk And High-Cost Patients". In: *Health Affairs*, 33(7), pp.1123-1131.

Beez, U., Humm, B.G. and Walsh, P. (2015), "Semantic AutoSuggest for Electronic Health Records", In: *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, 760-765.

Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. University of California. Available at: https://www.ics.uci.edu/~fielding/pubs/dissertation/ fielding_dissertation.pdf [Accessed 27 Jun. 2017].

Ferraiolo, D.F. and Kuhn, D.R. (1992). "Role Based Access Controls". In: *15th National Computer Security Conference*. pp.554 - 563.

Gomes, A., Ziviani, A., Correa, B., Teixeira, I. and Moreira, V. (2012). "SPLiCE". In: *Proceedings of the 2nd ACM SIGHIT symposium on International health informatics - IHI '12*. pp. 721-725.

Humm, B.G. and Walsh, P. (2015). "Flexible yet Efficient Management of Electronic Health Records". In: *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, 771-775.

Idelhauser, J., Beez, U., Humm, B.G., Walsh, P. (2016). "A Clinical Decision Support System for Personalized Medicine". In: *U. Bleimann, B. Humm, R. Loew, I. Stengel, P. Walsh (Eds): Proceedings of the 2016 European Collaborative Research Conference (CERC 2016)*, pp 132 - 145. Cork, Ireland, 2016. ISSN 2220 - 4164.

Jacobs, D. and Aulbach, S. (2007). "Ruminations on Multi-Tenant Databases". In: *BTW Proceedings, volume 103 of LNI*. Aachen Germany: GI, pp.514--521.

Kuo, A. (2011). "Opportunities and Challenges of Cloud Computing to Improve Health Care Services". In: *Journal of Medical Internet Research*, 13(3), p.e67.

ISO (2006). *International Organization for Standardization (ISO): Health informatics - HL7 version 3 - Reference information model - Release 1*. Available at: http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=40399.

Yamamoto, K., Sumi, E., Yamazaki, T., Asai, K., Yamori, M., Teramukai, S., Bessho, K., Yokode, M. and Fukushima, M. (2012). "A pragmatic method for electronic medical record-based observational studies: developing an electronic medical records retrieval system for clinical research". In: *BMJ Open*, 2(6), p.e001622.