Alignment of Business Models and Software: Using an Architecture-Centric Method to the Case of a Healthcare Information System

Marcos López-Sanz *marcos.lopez@urjc.es*
Kybele Research Group, Rey Juan Carlos University
Móstoles (Madrid), Spain

Valeria de Castro *valeria.decastro@urjc.es*
Kybele Research Group, Rey Juan Carlos University
Móstoles (Madrid), Spain

Esperanza Marcos *esperanza.marcos@urjc.es*
Kybele Research Group, Rey Juan Carlos University
Móstoles (Madrid), Spain

Abstract

The alignment of business issues with technological service-oriented solutions has proven to be a crucial aspect of modern business development. In this regard, the provision of methods to solve the gap between business and technology becomes absolutely necessary. This paper presents a proposal to systematize that leap by defining a development method centred on the concept of Architecture. The use of different architectural models at different levels of abstraction (along with the definition of model transformations between them) allows for the establishment of a trace between the business-level elements and software elements that are derived from them. Key benefits of our proposal are, on the one hand, the provision of a method for business-technology alignment and, on the other hand, the definition of a new model to represent the structure of a business. This proposal has been refined and validated using the case of an information system for the management of paediatric percentiles.

Keywords: Business-technology alignment, Architecture-centric development processes, service-orientation, Model-Driven Development.

1. Introduction

Many companies are moving towards an economic scheme based on the provision of services through the clear definition of the capabilities offered [7]. This scheme has evolved so that services are often presented as a key integral part for conducting businesses today [22]. In this service-oriented scenario, technology has become a necessary and profitable ally. Thus, during the past decade, service-orientation has spread to become one of the most important computing paradigms within modern business [11]. The service concept has therefore served to align the business with an appropriate software support [4]. In both areas (business and software), the identification of the participants and the nature of their relationships is crucial. In this regard, the Architecture can be used as an adequate artefact to progress from the business domain to the software one, using the service-orientation as paradigmatic base [14].

Using architectural models to represent both business and software architectures is one of the best approaches that can facilitate the alignment of business with an appropriate software support [10]. This approach is even more effective in service-oriented environments where the concept of service can be perfectly used to represent entities in both domains [2].

Following the previous reasoning, in this paper we present an architecture-centric and service-oriented method for the alignment of business models and software. The basic idea relies on the use of architectural models as the central artefact of a software development process. Furthermore, as already noted, our proposed alignment method uses a model-driven approach [6], i.e., it contains the definition of models and the specification of transformation rules between them. We defend that the use of model-based techniques, and more specifically the use of model transformations, can effectively reduce the gap between the definition of business services and their transformation into technological services [3].

In this paper we illustrate the feasibility of the proposal by showing its application to a case study from the real world. Such case study is based on the development of an information system for managing growth percentiles used by paediatricians to track their patient's development.

The remainder of this paper is organized as follows: Section 2 presents an analysis of some of the most relevant related works. Section 3 presents an overview of the method and the models included. In Section 4 the proposed alignment process is applied to a case study and, finally, Section 5 concludes the paper and shows some future work that arise from the research accomplished.

2. Related works

With regard to proposals that address the development of service-oriented (based on models or otherwise) solutions, we find many of them in the European Framework Programme. Projects like SeCSE [1] addressed the modelling and development of service-oriented solutions, but without considering the architectural modelling as a central aspect of the development process. Other projects such as SENSORIA [9] or PLASTIC [20] consider modelling services in technology, but again consider architecture as a mere additional modelling artefact.

Related to model-driven development of service-oriented software we also find the case of Zhang et al. [24] which, like the SCA's proposal [17], considers modelling software architectures based on components of services as modelling entities top level, but regardless of business processes or their characteristics. Perovich et al. [19] propose a development methodology that begins with the definition of CIM models for the architecture and ends up close to the code level. This proposal, however, does not include support for behavioural models or model transformations.

Finally, we would like to mention two initiatives from the commercial scope that address the development of a technological solution from the description of the business and which are focused on the architecture. On the one hand, Motion Lite [14], which enables the development of Web services but, although Microsoft claims that the defined process is traceable and "partially automated", there is no clear reference as to what their models or the transformation rules are. Moreover, IBM has defined CBM (Component Business Model) to define "business transformation, prioritizing the strategic objectives and linking solutions through traditional applications or SOA solutions" [24].

3. General description of the method and models included

As it is shown on the left side of Fig. 1, in our proposed method the Architecture is located in the centre of the development process. At the business-level a concrete model of the architecture reflecting the business structure is defined. The software architecture is defined at a lower level of abstraction, where it is related to other aspects of the software under development such as the interface or behaviour. Opposite, the right side of Fig. 1, shows the specific models involved in the proposal. This separation in models is done according to the abstraction levels defined by MDA [15] and includes the definition of a set of model transformation rules (not shown here for the sake of clarity).

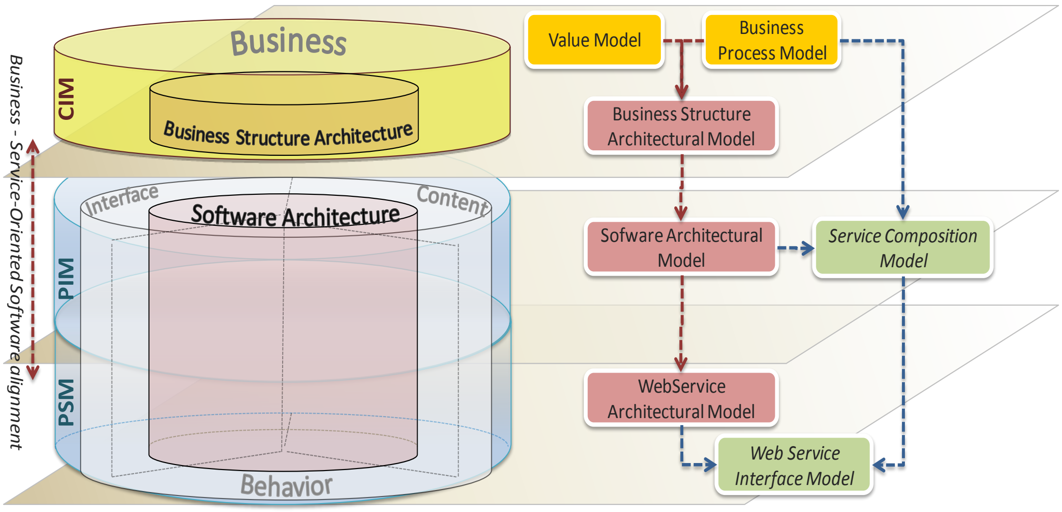

Fig. 1. Overview of the architecture-centric method.

In the following we briefly describe the models that we propose to represent, separately, the business context, the modelling of the architecture, and the behavioural aspect:

- **Modelling the business context**. The information reflected in this level refers to commercial activities and the exchange of value that occur as part of the business, as well as defining the specific business processes involved. The models proposed are [5](De Castro et al., 2011): a *value model*, which represents a set of values and exchange activities conducted by business actors; and a *business process model*, created to understand and describe business processes related to the environment in which the system is used. The notations used for those models are *e3value* [8] for the former model and BPMN [16] for the later.
- **Modelling the Architecture**. For modelling the Architecture we propose the definition of three different models: a *business structure architectural model* at the Computational Independent Model (CIM) level, that includes all the entities involved in the business and the relationships established among them (*contracts*); a *software architectural model* at the Platform Independent Model (PIM) level, which includes a reformulation and extension of the contents of the previous model but focused on conceptual services instead (PIM level); and, finally, a *Web Service architectural model* at the Platform Specific Model (PSM) level, containing information collected at PIM level and making use of the concepts related to Web services technologies [23]. All models are represented by extended UML class diagrams [12].
- **Modelling the behaviour**. Architectural models allow the identification of the structure of the software to implement. However, the specific behaviour needs to be modelled independently. For this purpose, the method presented in this paper defines three models: two at PIM level which are a *service composition model* and an *extended service composition model*, both of them represented with UML activity diagrams, and gathering information about the workflow necessary to carry out a service; and one at PSM level, *Web Service interface model*, that will be instanced as many times as required to describe the interface of every Web Service implemented in the system.

4. Business-Technology Alignment: Application to a Case Study

This section presents the set of steps containing the proposed alignment process illustrated by its application to a case study. In this paper we will only refer to the models corresponding to the architecture since its main objective is to show the benefits behind following an architecture-centric process. This way, we will show the business structure, the conceptual software and Web services models of the case study.

The case study we have used to validate the process is based on the development of an information system for the management of children percentiles at the Rey Juan Carlos Hospital (Móstoles, Spain) that seeks to improve the monitoring of its growth by both parents and by medical practitioners. It consists of a mobile application intended for parents, which allows visualizing and editing growth charts and immunization schedules among other features; and a desktop application for paediatricians, which can send notifications or request data synchronization percentile for better monitoring. Both the mobile and the desktop application communicate through a Web service-based underlying secured layer.

In the following subsections we explain each step given to leap from a description of the structure of the business scenario of the case study to the representation of the architecture of the implementation of the information system provided as solution to such working scenario.

4.1. Step 1: Obtaining the architecture of the business structure

The first step of the proposed method is to provide a complete representation of the business situation of the case study. To do this we start modelling all the services and values identified in the scope of the business scenario modelling a *business value model*. To obtain it, it was necessary to speak directly with the health professionals who indicated us the business needs to cover. For example, from the point of view of the paediatrician, it was required for the information system to create, the ability to synchronize data from different children. Another business activity mentioned was that, from the point of view of the parents taking advantage of part of the information system to be created, they wanted that it should be possible to graphically display the growth percentiles according to the standard WHO (*World Health Organization*) charts.

Knowing all these business needs allowed us to identify the services (understood as *business activities*) that the information system should provide to both paediatricians and parents (identified as *actors* in the business scenario). Alongside with this value model, we also proposed to model the characteristics of each of the processes involved in this business scenario. This was done through the creation of a *business process model* in BPMN notation.

However, and according to the main aim of this paper, the first set of model transformations that define our process refers to the objective of obtaining a *business structure architectural model*. This was done using as source both the *value model* and the *business process models*. The result is the model shown in Fig. 2.

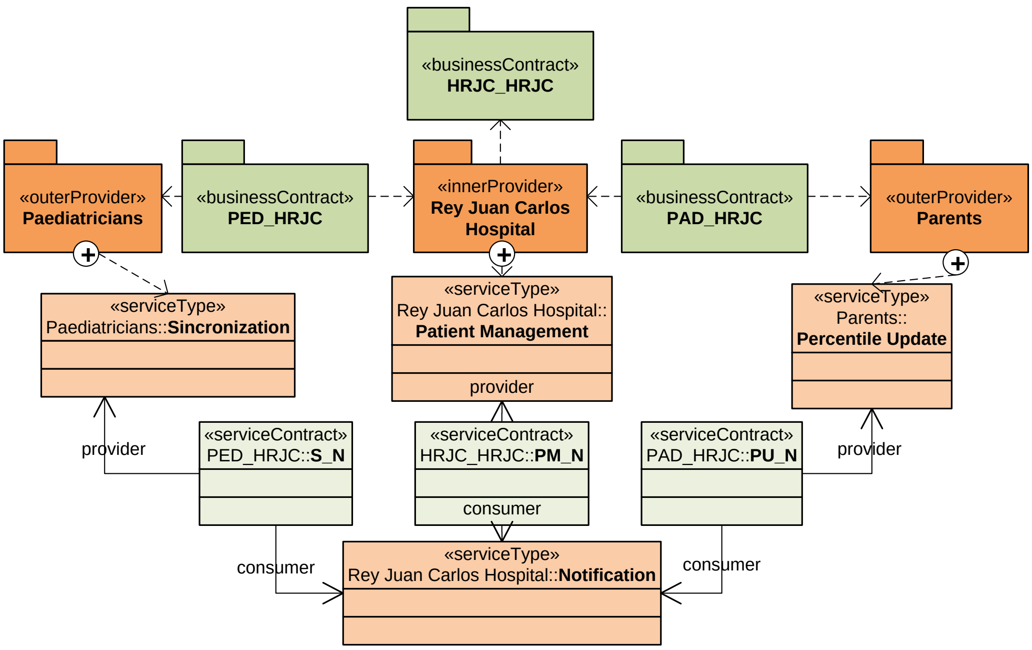

Fig. 2. Business structure architectural model.

In Fig. 2, business entities (*parents* and *paediatricians*) have become service providers (*inner/outerProvider*). The architectural element that acts as a connector in this model is the business contract (*businessContract*). This element appears whenever an object of value is exchanged between two actors in the *value model*. Although service providers are entities that take the responsibility of the business situation, who actually perform the business activities are the services. As such, value activities will be transformed directly into services (*serviceType*). To illustrate the importance of having defined previously an additional (and independent) business process model, it is possible to discuss how the activity "percentile update" that exists in such model (not shown here) is used as a source to include in the architectural model a particular service which is responsible for carrying out such tasks.

4.2. Step 2: Obtaining the software architecture

Once the business structure has been modelled, our method facilitates the progression in the development of the system. To achieve that we take as input the *business structure architectural model* and a set of transformation rules to obtain a preliminary version of the software architecture. In parallel, the service composition model corresponding to the behavioural aspect of the system to create is also required. The *software architectural model* for the case study is shown in Fig. 3. This is obtained by applying model transformation rules associated to the proposed method as well as the already mentioned behavioural models (not included here).

The core elements of this model represent the conceptual structure of the software to implement. The origin of each is as follows: *services* and *service types* are derived from business services modelled in the *business structure architectural model*. For instance, for the case study, the service requesting the information recorded by the parents in their mobile application has been transformed directly into a software service (PAD_FrontEnd).

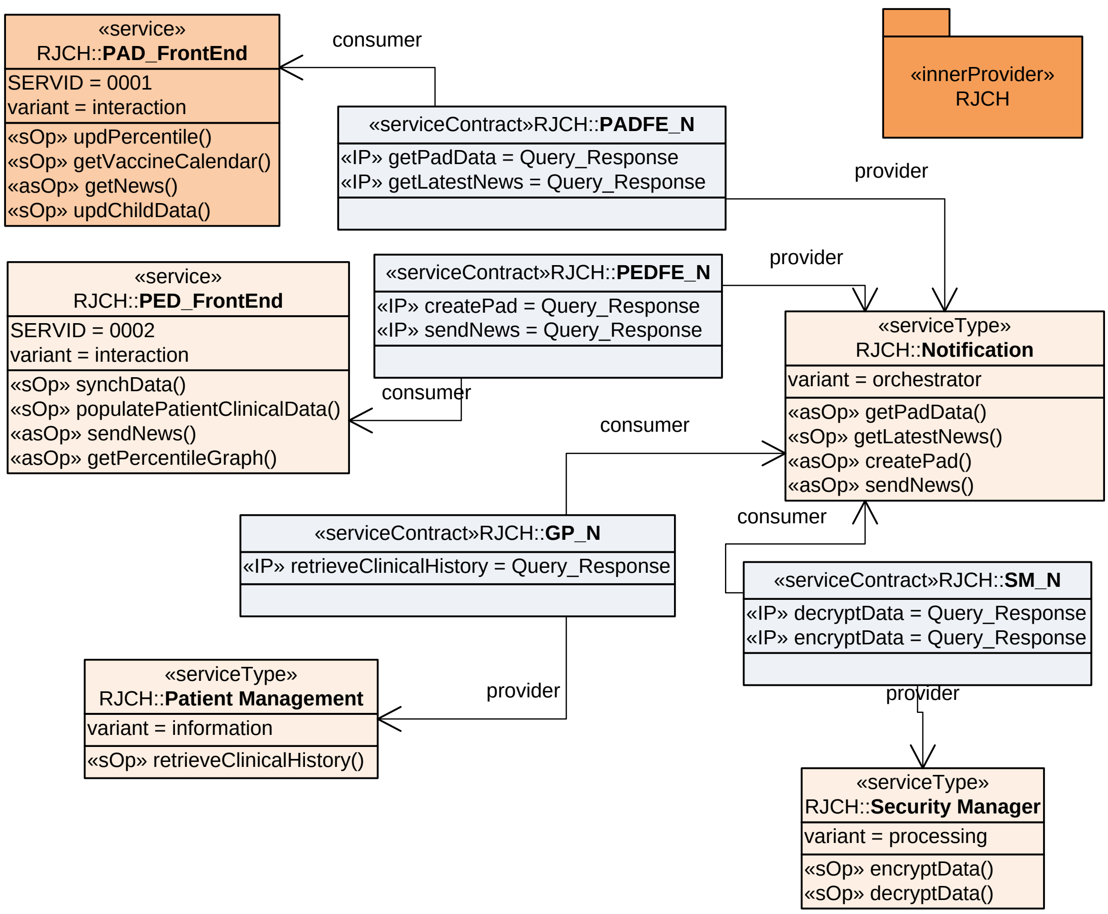

Fig. 3. Software architectural model at PIM level.

The model shown in Fig. 3 also contains other services that do not come directly from the business architecture (such as the *SecurityManager*), but depend on the architectural design chosen by the developers. As in the aforementioned case of services, we defined transformations to specify the relationships between the different conceptual services (*service contracts*), service operations (understood as individual functional units associated with each service, which in this case we obtain from the service composition model which in turn are obtained from the business process model) and the message exchange patterns established between services.

4.3. Step 3: Obtaining the Web Services Architecture

The final architectural model is the one that represents the elements that will be implemented as software artefacts and according to the characteristics of the Web Services technologies. The conceptual architectural elements are transformed into entities from which it will be possible to generate code snippets and service interfaces. The resulting Web Service architectural model for the case study is shown in Fig. 4.

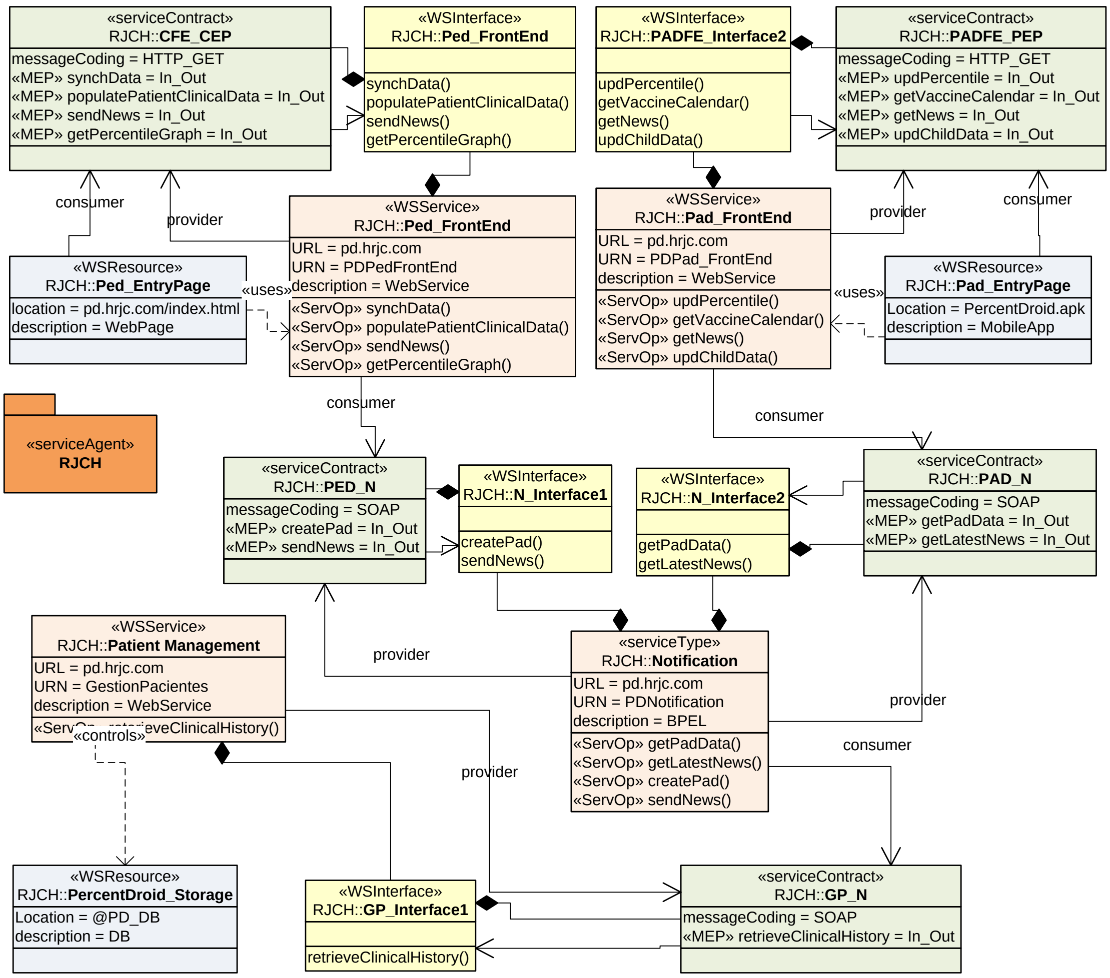

Fig. 4. Web Service Architectural model of the PSM level.

The transformation rules applied to obtain the model in Fig. 4 revolve around the definition of the Web service as main architectural element. This way, in our case study for example, the conceptual service that synchronizes percentiles (*PED_FrontEnd*), which can be traced to (and are therefore aligned with) the value activity initially modelled at business level, is automatically transformed into a Web service. Other elements that are semiautomatically obtained are, for instance, the resources managed by the services (*WSResource*), derived from service type defined at the conceptual level (*processing*, *information*, *interaction*, etc.) or service operations (*ServOp*) as atomic service capabilities derived directly from the service operations defined at the level of the conceptual architecture (step 2).

4.4. The front-end application for Android

The previous architectural model serves the developers as starting point to implement several elements of the proposed information system. For example, the mobile app that is given to parents to keep track of the growing of their children is the result programming an Android app derived from the existence of the architectural element *Pad_EntryPage*, identified as WSResource (and given the value "MobileApp" to its description attribute).

This app acts as front-end resource for the information system developed from the parents' perspective. It is not in the scope of the current work to show the modelling of mobile interfaces which is a work currently under development to be included as part of the whole alignment framework.

In Fig.5 it is possible to see an overview of the working interface developed for this mobile app (*PercentDroid*). Texts in the screenshots are in Spanish since the intended users are currently restricted to Spain.

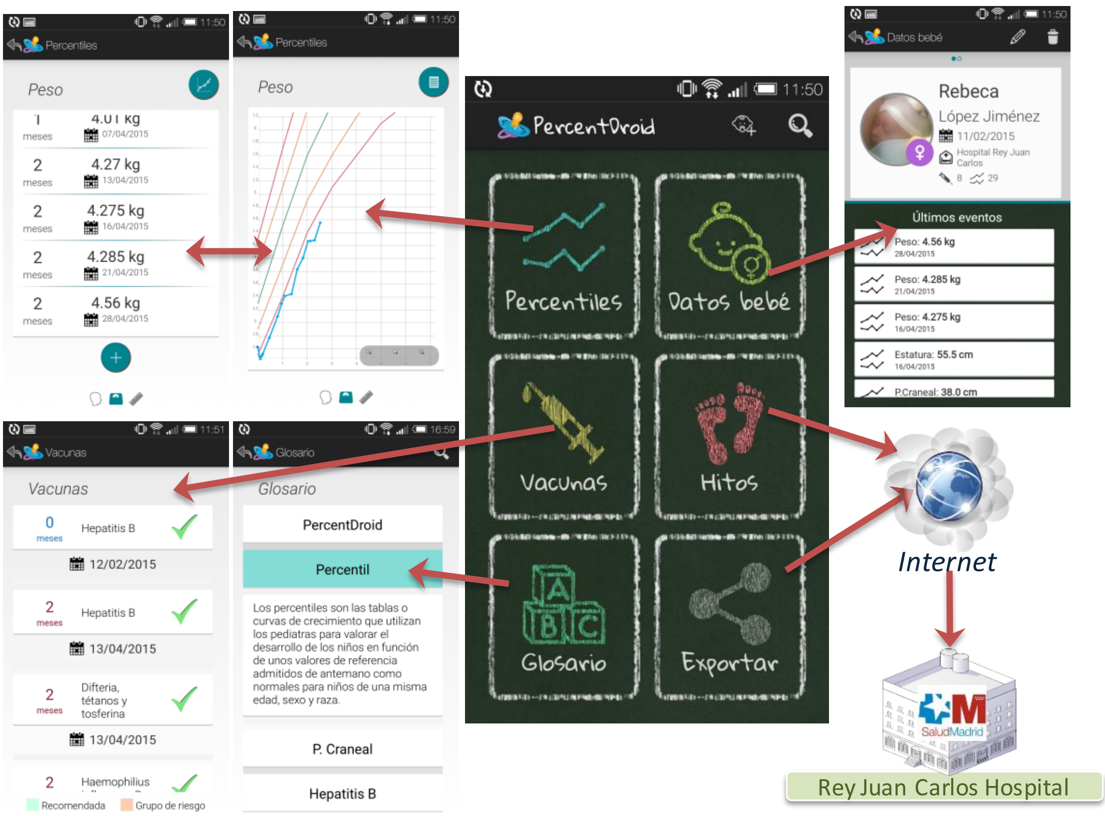

Fig. 5. Overview of the *PercentDroid* mobile app.

4.5. Lessons learned

The first lesson we have obtained from the application of the proposed method to the case presented is that the use of a specific architectural model for the business structure eases tackling the problem of alignment between business and software. The defined high-level models contain elements that can be easily transformed into software architectural elements. After the application of the proposal we have confirmed that the alignment is much easier than with traditional development methods since the problem can be reduced to an alignment between architectural models.

In this sense, the decision to use a service-oriented approach to software development may also increase the business alignment with the software thanks to the use of service-orientation as shared paradigm and vocabulary. When understood from a perspective of value activities and value exchanges, it is easier to align a business scenario with a service-oriented software architecture than with other architectures following different paradigms. Service orientation has particular characteristics that are much closer to the business concepts so it is possible to represent much easier the any underlying supporting software [11]. As stated in [18] service-orientation creates service level abstractions that correspond with how a business really works.

Finally, the use of model transformations makes explicit the alignment of business and software. The definition of transformations between elements of different models adds other important advantages to this process, such as the ability to manage traceability relationships between the high-level business processes and the service-oriented software systems that support them [21].

5. Conclusions and future lines of research

In this paper we have presented an architecture-centric and model-driven method in order to address the challenge of aligning business processes with software solutions through service orientation. The fact of considering the architecture as a central artefact of the alignment process allows us to recognize, quite early in the development process, the key elements of the business structure and their interactions thanks of the concept of service. Using a model-based approach for this alignment process further reduces the complexity because it allows the business elements specified in business models to be converted into increasingly detailed (and near the target software) architecture elements. Thus, the structural information is transferred from the business scope to the software layers via service-oriented architectural models. In this regard, as an additional contribution of this work, we have described the definition of a new business model: the *business structure architectural model*. This model offers a new vision of business that complements other well-known models such as the *value model* or the *business process model*.

The proposed process has been refined and validated through its application to a real case study of the healthcare sector. The objective was to help managing the information about the development and growth of paediatric patients. Despite all the benefits and features of our proposal, we are working on several lines of research associated with the defined process. On the one hand, we are working to complete the development of a set of modelling toolkits that support the proposed architecture-centric method. On the other hand, we would also like to note that we are currently applying the method presented in some other case studies in the field of healthcare services in cardiology.

Acknowledgements

This research has been partially funded by the Regional Government of Madrid under the SICOMORo-CM (S2013/ICE-3006) project, by the MASAI (TIN-2011-22617) and ELASTIC (TIN2014-52938-C2-1-R) projects, financed by the Spanish Ministry of Science and Innovation, and by the Service Science, Management and Engineering-GES2ME Research Excellence Group (Ref. 30VCPIGI05) co-funded by Rey Juan Carlos University and Banco Santander. We would also like to thank Dr. Myriam Herrero and Dr. Erika Jimenez of the Rey Juan Carlos Hospital, for their assistance in the definition of case study.

References

[1]. ATOS, SeCSE methodology, version 3, IST European integrated project SeCSE, in: Proceedings of Sixth Framework Programme, Deliverable A5.D4.2.
[2]. Aversano, L., Grasso, C., Tortorella, M., A Literature Review of Business/IT Alignment Strategies, Procedia Technology, 5, pp. 462-474, ISSN 2212-0173.
[3]. Broy M. Model Driven, Architecture-Centric Modeling in Software Development. In Proc. of 9th Intl. Conf. in Eng. Complex Comp. Syst., pp. 3-12, IEEE Computer Society, 2004.
[4]. Cherbakov, I., Galambos, G., Harishankar, R., Kalyana, S., Rackham, G, 2005. Impact of Service Orientation at Business Level. IBM Systems Journal, vol. 44, issue 4, pp. 653-668.
[5]. De Castro, V., Marcos, E., Vara, J. M. Applying CIM-to-PIM model transformations for the service-oriented development of information systems. IST 53(1): 87-105. 2011.
[6]. De Castro, V., Marcos, E., Wieringa, R., 2009. Towards a Service-oriented MDA-Based Approach to the Alignment of Business Processes with it Systems: From the Business Model to a WS Composition Model. Int J on Coop. Inf. Syst.. 18(2): 225-260.
[7]. Glissmann S., Sanz, J.L.C., 2010. Business Architectures for the Design of Enterprise Service Systems. Handbook of Service Science, Springer.
[8]. Gordijn, J., Akkermans, J.M., 2003. Value based requirements engineering: exploring innovative e-commerce idea, Requirements Engineering Journal 8 (2) 114–134.
[9]. J.L. Fiadeiro, A. Lopes, L. Bocchi, The SENSORIA Reference Modelling Language: Primitives for Service Description, available at www.sensoria-ist.edu, 2006.
[10]. Karakostas, B., Y. Zorgios, 2008. Engineering Service Oriented Systems: A Model Driven Approach. IGI Global, Hershey.
[11]. Krafzig, D., Banke, K., Slama, D., 2004. Enterprise SOA Service Oriented Architecture Best Practices, Prentice Hall PTR, Upper Saddle River.
[12]. Marcos, E., De Castro, V., Vela, B., 2003. Representing web services with UML: a case study. 1st Int. Conf. on Service Oriented Computing, LNCS 2910, 2003, pp. 17–27.
[13]. Merrifield R., Tobey, J., 2006. Motion Lite: A Rapid Application of the Business Architecture Techniques Used by Microsoft Motion. Microsoft.
[14]. Nayak, N., Linehan, M. H., Nigam, A., et al., 2007. Core business architecture for a service-oriented enterprise. IBM Systems Journal 46(4): 723-742.
[15]. OMG, 2001. Model Driven Architecture (MDA), Document No. ormsc/2001-07-01, available at: http://www.omg.com/mda.
[16]. OMG, 2008. Business Process Modelling Notation, Version 1.1, http://www.bpmn.org/Documents/BPMN1-1Specification.pdf
[17]. Open Service Oriented Architecture Collaboration, Service Component Architecture Project. http://www.osoa.org/ , 2007.
[18]. Papazoglou, M., Traverso, P., Dustdar, S., Leymann, F., 2008. Service-Oriented Computing. Research Roadmap. Int. J. of Coop. Inf. Syst., 17 (2), pp. 223–255.
[19]. Perovich, D., Bastarrica, M.C., Rojas, C., 2009. Model-driven approach to software architecture design. Proc. of the 4th Int. Workshop on Sharing and Reusing Architectural Knowledge, SHARK'09, IEEE Computer Society, Vancouver, Canada, 2009, pp. 1–8.
[20]. PLASTIC Project. (2006). D1.2: Formal description of the PLASTIC conceptual model and of its relationship with the PLASTIC platform toolset. http://www-c.inria.fr/plastic
[21]. Santiago, I., Jiménez, A., Vara, J.M., De Castro, V., Bollati, V., Marcos, E., 2012. Model-Driven Engineering as a new landscape for traceability management: A systematic literature review. Information & Software Technology 54(12): 1340-1356.

[22]. Sanz, J.L.C., Ren, G., Glissmann, S., Spohrer, J., Leung, Y.T., 2010. What is Business Architecture and how can it contribute to Service Science. Frontiers in Service Conference, Karlstad, Sweden.
[23]. W3C, 2004. Web Services Architecture (WSA). http://www.w3.org/TR/ws-arch/.
[24]. Zhang T., S. Ying, S. Cao, X. Jia. A Modeling Framework for Service-Oriented Architecture, in: Proceedings of the Sixth International Conference on Quality Software (QSIC 2006), pp. 219-226
[25]. Zhang, L.-J., Zhou, N., Chee, Y.-M., Jalaldeen, A., Ponnalagu, K., Sindhgatta, R., Arsanjani, A., Bernardini, F., 2008.SOMA-ME: A platform for the model-driven design of SOA solutions. IBM Systems Journal 47(3): 397-413.