Requirements Engineering and Business Process Management as preconditions for the application of the Cloud Blueprinting Model

Lisardo Prieto-González[1], Johannes Schubert[2]

Abstract: In this paper, we test the cloud blueprinting approach against a modelled use case. Hereby, we show the importance of both Requirements Engineering (RE) and Business Process Management (BPM) to that application. We highlight the main features of Business-Process-as-a-Service (BPaaS) and the Cloud Blueprint approach, and explain how the integration of requirements following a standard classification allows to enhance the interoperability with other existing models. This is done by means of a business process example modelled with BPMN and later translated to the Blueprint Specification Language (BSL) with the help of RE.

Keywords: Requirements Engineering, Business Process Management, IT, Cloud Computing, Software Engineering, Cloud Blueprinting Model.

1 Introduction

As automation of internal processes is one of the main concerns for businesses operating at different scales, the advent of cloud computing could make it possible for cloud users to outsource the automation processes once undertaken in-house. Business-Process-as-a-Service (BPaaS) is an extension [Hö09] [Zh12] [Mo11] to the traditional three-level cloud stack [Va08], and is already promoted for the utilization of cloud computing in different scenarios [SSY14]. With BPaaS, business processes can be delivered via cloud either vertically or horizontally [LBM12].

The aim of this paper is to test the cloud blueprinting approach against a modelled use case and hereby show the importance of both Requirements Engineering (RE) and Business Process Management (BPM) to that application. The paper is structured as follows: in section 2, we highlight the main features of BPaaS and the cloud blueprint approach. In section 3, we propose a standard classification for RE. In section 4, we briefly explain a given use case in a health care scenario which is modelled with BPMN and its potential for the BPaaS layer. One process is exemplary then translated to the blueprint specification language (BSL) with the help of RE. In section 5, we give a conclusion and encourage future work on this promising approach.

[1] Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganés (Madrid) Spain, lisardo.prieto@uc3m.es
[2] SRH Hochschule Berlin, Institute of Information System, Ernst-Reuter-Platz 10, 10587 Berlin, johannes.schubert@srh-hochschule-berlin.de

2 BPaaS and Cloud Blueprinting

2.1 Business-process-as-a-Service

Although slightly mentioned before [Ac11], the BPaaS concept was more frequently discussed from 2011 when Papazoglou and van den Heuvel explained their "*syndicated multichannel cloud delivery model*" [PH11] adding the BPaaS layer to the traditional cloud stack consisting of SaaS, PaaS, and IaaS [MG09]. Purpose of the BPaaS model is a full integration of the other layers to a cloud ecosystem based on the specific needs which are given by the business processes of the cloud user. Central focal point of the BPaaS approach is the virtualization of end-to-end processes which are translated into the proposed cloud blueprinting languages in order to standardize deployment plans and adapt cloud configuration according to the needs of the respective business processes [PH11]. Main advantage of the BPaaS paradigm is the change from "*a monolithic SaaS/PaaS/IaaS stack architecture where a one-size-fits-all mentality prevails*" [Ng12] to a mash up of cloud services in order to provide tailored service-based applications (SBA).

2.2 The Cloud Blueprint Model

The blueprint languages are proposed in order to apply the BPaaS model to the cloud. In other words, based on given business processes, standardized requirement templates for SaaS, PaaS, and IaaS can be filled accordingly [PH11]. Even though the model is very mature it is still at system design level and use cases are rarely to find [CWW15]. While BPaaS is becoming more popular, research on the blueprint model only reveals literature related to the "4CaaSt" FP7 project [Ga12], where the approach has been developed. The blueprint model is the projects' reaction to at least seven lacks the researchers could find in a state-of-the-art study on cloud standardization support [Ta12]. In [Ng11], the authors present the blueprint as an abstract description of a cloud service offer hiding the technical details and complexities. The blueprint template is more elaborated in [Ng12], explaining the structural schema of the blueprint XML file, providing more details about the proposed language. The fundamental blueprint structure is described as follows:

Basic Properties: basic description properties (e.g. unique ID, release date),

Offering: properties of according cloud service offerings (e.g. names, functionalities, signature interfaces, interaction protocols),

Implementation Artifacts: list of artifacts implementing the offerings (e.g. binary files, configuration files, deployable web packages).

Resource Requirements: required cloud resources including their QoS requirements,

Virtual Architecture: visualization of the target architecture to provide interdependencies and interconnections between the offerings, implementation artifacts and resource requirements,

Policy: non-functional properties as declared in service level agreements (e.g. security, privacy, compliance requirements).

The most detailed description of the blueprint model is given in [Ng13]. Here, the term Blueprint Specification Language (BSL) was introduced, displacing and extending the previously used Blueprint Definition Language (BDL). The blueprint structure became more diverse with the help of modules that reduce complexity and focus on specified requirements.

Finally, Nguyen [Ng13] gives an overview on future issues which are e.g. a potential extension of categories, resource description modules, and further validation of the concept. Concluding the findings about BPaaS and the blueprint approach, we can make the following general statements:

- The BPaaS paradigm supports the definition and deployment of business process related cloud services.
- The blueprint model is a solid and stable approach to apply the BPaaS layer to any kind of cloud marketplaces.
- The Blueprint Specification Language is supporting RE with a standardized structure and extensible parameter set.

3 Standard classification for RE

Requirements capture and analysis phase is used in the different methodologies to acquire the customer's needs and translate them into a set of requirements that define what the system must do and how well it must perform among other characteristics. During this phase, the systems engineer must ensure that the requirements present various attributes [Li01] [Ma94]. While business process modelling focuses on the interdependency of tasks and users within the business process, it creates a managerial perspective on the set of tasks. The BSL template, however, is supposed to integrate different system offers to match with the requested task. Therefore, specifying requirements by using standardized classifications is an inevitable precondition to let the cloud blueprinting concept take full effect.

Summing up, requirements must be, among other set of characteristics: complete, comprehensive, concise, unambiguous and understandable [SBA13].

Usually, requirements are documented following a standard classification. Although there are general classification schemes [Li01], this study will be focused in the categories defined under FURPS+ [Gr92] for Information Systems by Hewlett-Packard. FURPS+ Is the acronym for Functional, Usability, Reliability, Performance, Supportability and the extension (+) includes three additional categories: Constraints, Interface Requirements and Business Rules. In the following, the categories and sub-categories are shown:

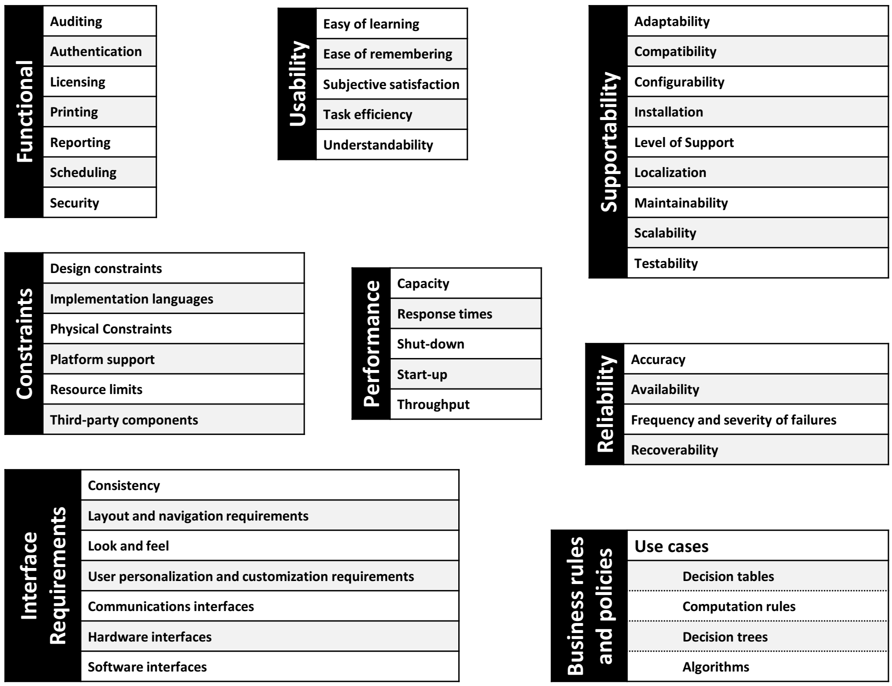

Figure 1. Requirements categories defined by FURPS+

In order to verify that the different types of system-wide requirements covered by FURPS+ have been considered, it seems to be useful to follow an organized strategy during the capture phase and use a checklist [Ec12].

4 Translation of a business process to BSL

Since the blueprint model is intended to organize and structure requirements and definitions for service-based applications (SBA), we are seeking to evaluate its benefits with applying the model to a given use case in a health care scenario. The use case is present in BPMN which will be translated to the blueprint model. The translation is done in two steps. First, the relevant processes modelled in BPMN will be extracted. Second, the core definitions of these processes are applied to the blueprint template. Actually, the business processes imply all technical requirements directly, so the target blueprint will remain incomplete at this stage. However, in this paper we are seeking for the general application, so that this issue is not considered now.

Within our use case, we introduce a smart blister dispenser for residential care facilities. The blister dispenser controls whenever a medication blister was taken by the occupant of the care facility and checks against a database whether medication needs to be taken or not. The complete process with all users involved is modelled as shown in Figure 2.

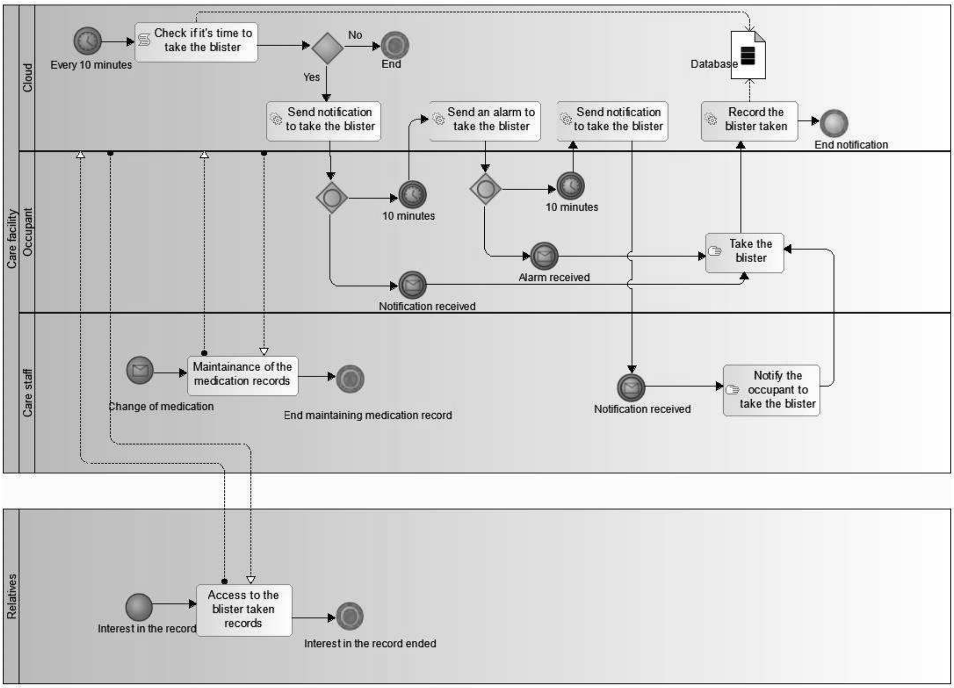

Figure 2. Full BPMN model for the smart blister dispenser use case

In order to create the BPaaS layer for this use case, the modelled processes in the cloud lane are extracted. In other words, the future business processes fulfilled by the cloud, are filtered and briefly explained. These processes and their properties are defined in Table 1.

Process	**User**	**Usage**	**Parameters**	**User Requirements**
Check for blister to be taken	Web back-end	Every 10 minutes match current time against time the blister needed to be taken by specific person stored in a database table	Request: current time Response: Person, dedicated output device, number of blisters, time of blister to be taken, status of blister taking	Server is able to handle script tasks on a regularly basis
Send notification for the blister to be taken	Server	If time of blister to be taken is less than 10 minutes compared to current time and blister was not taken yet, send an XML or JSON file to the dedicated end user device	Request: message status, number of blisters to be taken Response: status feedback	Server is able to push XML or JSON files
Send message for the blister to be taken	Server	Send an XML or JSON message to specific entity accordingly to the following conditions: • To the dedicated end-user device, if time of blister to be taken is between 10 and 20 minutes compared to current time and blister was not taken yet. Notification. • To the nurse/nursing manager logged in, if time of blister to be taken is more than 20 minutes and less than 30 minutes compared to current time and blister was not taken yet. Alarm.	Request: message status, number of blisters to be taken Response: status feedback	Server is able to push XML or JSON files
Log blister taking	Server	After receiving an XML (?) message about the usage of a specific blister, insert the associated data to the database	Request: current time, medication time, person, number of blisters taken Response: status feedback	Server is able to handle incoming XML or JSON messages
Point out blister taking to the occupant	Web front-end	If XML file (?) arrives at the nurses' computer having the web front-end connected to the cloud web service, a toaster in the browser and a desktop notification open to provide information	Request: message status, number of blisters to be taken Response: feedback	Smart Watch is attached to the occupant

Table 1: Example process extracted from use-case

The human readable table of business processes can now be translated to target blueprints. As follows, we exemplary apply the first process to the most elaborated blueprint XSD template explained in [Ng13] and the XML implementation explained in [Ng12]. A brief implementation example is presented in Figure 3.

```
BlisterUsageCheck-CoreBlueprint.xml
<basic_information_section>
        <blueprintID>1</blueprintID>
        <blueprintName>Check for blister to be taken</blueprintName>
        <description> Every 10 minutes match current time against time
        the blister needed to be taken by specific person stored in a
        database table </description>
        <ownership>OpSIT</ownership>
        <status>unresolved</status>
</basic_information_section>
<requirements_section>
        <Requirement>
                <requirementID>1</requirementID>
                <requirementName>Regular script handling</requirementName>
                <requirementType>SaaS</requirementType>
        </Requirement>
        <Requirement>
                <requirementID>2</requirementID>
                <requirementName>Database</requirementName>
                <requirementType>PaaS</requirementType>
        </Requirement>
</requirements_section>
```

Figure 3: BSL core module for the blister usage check

By means of compilers theory, it would be feasible to develop a converter, able transform directly from BPMN notation to BSL language. However, even if it seems a trivial task, it requires some advanced semantic artificial intelligence processing, plus a large and dynamically growing and structured dataset of specific requirements associated to certain actions/interactions/cases. This dataset can be designed to grow by the application of semi-automatic machine learning techniques, being required human experts in the initial enrichment phases, but becoming autonomous eventually. This way, the idea of a translator able to interpret the semantics of BPMN could check the cited dataset to infer the most promising requirements according to the particular cases, outcoming automatically a proper BSL description. Nevertheless, this approach is strongly dependent on the quality, complexity and completeness of the requirements dataset.

In this work we do not want to focus in the idea of a translator. Besides, we have considered its implementation in a later work, as we think it may improve a lot the conversion process, and can enrich the knowledge about the semantics between BPMN, RE and BSL.

Due to the very flexible structure of the blueprint model, we hereby confirm the application of business processes extracted from BPMN models to the blueprint XSD template. Using the XSD template obviously demands a precise definition of requirements. Summarized, our translation approach shows the importance of a clear definition of processes and its requirements already at the modelling level.

5 Conclusions and future work

In this paper, we highlighted the main features of BSL, and proposed a standard classification for RE. In order to create a sample blueprint for a given use case, we explained how the business processes modelled in BPMN can be translated to a Cloud Blueprint. With a simple example, we could emphasize previous statements which explain that the extensible design of the blueprint template is able to include more requirements for the engineering of service-based applications [Ng12] [Ng13]. Furthermore, we strongly recommend the integration of requirements following a standard classification as this allows to enhance the interoperability with other existing models, increasing the semantic concept of the solution, and reducing potential risks derived from an unstructured analytics development phase. Also, we believe that such integration may lead to much more engagement in the scientific community on the cloud blueprint model, which makes it even more applicable for cloud marketplaces, developers and service providers.

Acknowledgements

This publication is based on the research project “OpSIT” which received financial funding from the ministry of education, science, research and technology (BMBF) under the funding sign 16SV6048 within the programme „IKT 2020 - Forschung für Innovationen“. The authors are responsible for the content.

Bibliography

[Ac11] Accorsi, R.: Business process as a service: Chances for remote auditing. In: *Computer Software and Applications Conference Workshops (COMPSACW) IEEE 35th Annual*. 2011.. S. 398-403.

[CWW15] Chang, V.; Walters, R.J.; Wills, G.: Cloud Computing and Frameworks for Organisational Cloud Adoption. *Delivery and Adoption of Cloud Computing Services in Contemporary Organizations* (1), 2015.

[Ec12] Eclipse Process Framework: *Checklist: System-Wide Requirements (FURPS+)*. http://epf.eclipse.org/wikis/openup/core.tech.common.extend_supp/guidances/checklists/system_wide_rqmts_furps_3158BF2F.html, 2012.

[Ga12] Garcia-Gomez, S.: 4CaaSt: Comprehensive Management of Cloud Services through a PaaS. In: *10th International Symposium on Parallel and Distributed Processing with Applications*. 2012.. S. 494-499.

[Gr92] Grady, R.B.: *Practical software metrics for project management and process improvement*. Prentice-Hall, 1992.

[Hö09] Höing, A.: An orchestration as a service infrastructure using grid technologies and WS-BPEL. In *Service-Oriented Computing*. Berlin Heidelberg: Springer. S. 301-315, 2009.

[LBM12] Lau, C.; Birsan, V.; Madapusi, B.: *IBM Developer Works*. IBM. http://www.ibm.com/developerworks/websphere/library/techarticles/1203_lau/1203_lau.html, 2012. zuletzt geprüft am 06.08.2015.

[Li01] Lightsey, B.: *Systems engineering fundamentals*. (Report) DTIC., 2001.

[Ma94] Mazza, C. et al.: *Software engineering standards*. Prentice-Hall, 1994.

[MG09] Mell, P.; Grance, T.: The NIST definition of cloud computing. *Natl. Inst. Stand. Technol*, 53, 6 (2009), 50.

[Mo11] Moscato, F.: An analysis of mosaic ontology for cloud resources annotation. In: *Federated Conference on Computer Science and Information Systems*. 2011.. S. 973-980.

[Ng11] Nguyen, D.K.: Blueprint template support for engineering cloud-based services. In *Towards a Service-Based Internet*. Berlin Heidelberg: Springer. S. 26-37, 2011.

[Ng12] Nguyen, D.K.: Blueprinting Approach in Support of Cloud Computing. *Future Internet*, 4. S. 322-346, 2012.

[Ng13] Nguyen, D.K.: *Blueprint model and language for engineering cloud applications*. (Dissertation) Tilburg: CentER, Center for Economic Research., 2013.

[PH11] Papazoglou, M.P.; van den Heuvel, J.W.: Blueprinting the cloud. *IEEE Internet Computing* (6). S. 74-79, 2011.

[SBA13] Saavedra, R.; Ballejos, L.; Ale, M.: Software Requirements Quality Evaluation: State of the art and research challenges. In: *14th Argentine Symposium on Software Engineering*. 2013..

[SSY14] Sun, Y.; Su, J.; Yang, J.: *Separating execution and data management: a key to Business-Process-as-a-Service (BPaaS)*., 2014.

[Ta12] Taher, Y.: On engineering cloud applications-state of the art, shortcomings analysis, and approach. *Scalable Computing: Practice and Experience*, 3 (13). S. 215-231, 2012.

[Va08] Vaquero, M.: A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 1 (39). S. 50-55, 2008.

[Zh12] Zhang, L.J.: Editorial: Big services era: Global trends of cloud computing and big data. *IEEE Transactions on Services Computing* (4). S. 467-468, 2012.