Model-Driven Engineering meets the Platform-as-a-Service Model

Adrián Juan-Verdejo

Information Systems Chair
Stuttgart University, Germany
adrianppg@gmail.com

Abstract. Organisations who want to migrate their applications to Platform-as-a-Service (PaaS) environments have a lot of PaaS vendors to chose from and many PaaS offerings within them. At the PaaS abstraction level within the cloud service model, each vendor offers many different configurations to meet the specific organisation's criteria to migrate to PaaS and the application's requirements. PaaS providers and their offerings differ in terms of the supplied execution environments, databases, web servers, or development tools. Enterprises have difficulties to choose the right PaaS offering for them to deploy their applications to and have to manually configure their applications according to the PaaS selection. All in all, this is an error-prone process which model-driven approaches can facilitate. This paper shows the meta models needed to describe the application to be deployed to a PaaS offering, the PaaS offering itself, and the migrated applications with re-scattered components to both the local premises and PaaS. The InCLOUDer decision support system uses these meta models to support the deployment of the application to the PaaS offering which best helps enterprises in achieving the criteria which motivate them to start using PaaS environments. InCLOUDer keeps some of the application's components locally to fulfil the constraints of organisations related to security, data privacy, performance, compatibility, and governance.

1 Introduction

By deploying their applications to PaaS, today's software organisations—and small and medium enterprises (SMEs) in particular—can achieve key competitive advantages due to their business agility, how quickly they market the products they offer, and their ability to reach their potential users. PaaS environments have the potential to let organisations increase their abilities to scale up and down—by adding or removing resources to a node in use—, or in and out—by adding or removing nodes to a system—; cut their costs; and shorten the time they spend to market their software solutions. PaaS offerings hide low-level details from the developer, such as the underlying infrastructure including load balancers, virtual machines, servers, data storage and access, and networks. The PaaS developing environment frees developers from software versioning or resource provision and lets them use extra built-in services and capabilities and write code faster. PaaS vendors provide development tools, libraries, and application programming interfaces (APIs) to let developers access their storage resources and deploy their applications to their runtime environment and execute them. Due to the faster PaaS application testing and deployment; development teams easily try different configurations of their applications to test them, assess their applications' performance, and identify potential compatibility issues [1]. From a business perspective, as enterprises outsource computation to PaaS vendors, they can focus on their core business interests and do it better [2].

Problem statement. As a result of the potential advantages offered by PaaS, the demand of PaaS environments has been increasing for the last years and the number of PaaS providers accordingly. A myriad of vendors supply many PaaS offerings at various levels of abstraction with different capabilities, runtime environments, services, and supported platforms. The huge amount of PaaS vendors together with the various PaaS offerings with many configurable parts makes for a lot of options for organisations to chose from. So many options that organisations find difficulties in selecting the right PaaS offering for their software system requirements and the goals they want to achieve by using PaaS environments. These goals range from achieving easier extensibility by using the services offered in PaaS environments to easing the deployment, including improving the application's security and portability. Enterprises can choose from PaaS offerings with various security protocols, QoS guarantees, trust issues, tools form application migration and porting, followed standards, and cost models. SMEs' PaaS migration criteria might lead them to want to select the most reliable, performant, well-reputed, or cost-efficient offering [3]. Likewise, they might simply be interested in using the one which meets their technical requirements without much adaptation or developing effort. It is not obvious how to compare PaaS offerings according to these criteria or dimensions [4]. Additionally, it is not evident how to identify the potential of a PaaS offering over another one, especially on the long term. There is a lack of standards and metrics to identify differences and compare PaaS offerings [5].

The six dimensions identified in [4] to compare PaaS offerings are not independent. As a result some dimensions or sub-dimensions oppose others in some cases. Organisations need assistance to incorporate all this knowledge into the decision-making process involved in migrating applications to PaaS. Additionally, enterprises may face a well-known problem associated with the PaaS model, the vendor lock-in problem when they want to switch from one PaaS vendor to another one. The inability of some PaaS vendors to comply to standards hinders the application and data portability. But even if portability is supported in theory, the high complexity of the process if vendors do not facilitate it and the switching costs still discourage organisations from porting their applications [6]. Organisations want to be able to switch easily and transparently between PaaS providers whenever they need. They might want to move to a different provider due to a Service Level Agreement breach (SLA), a cost increase, or simply because the vendor went out of business. At the same time, organisations do not want to set their data and applications at risk in case they have to port their applications to a different PaaS environment. Typical risky scenarios include data loss or inability to run the application on a different platform.

Related work. The migration of systems and applications to cloud-enabled environments attracts a lot of attention from the research community [7] [3]. The REMICS FP7 project [8] reuses and migrates legacy applications to interoperable cloud services by using concepts from the Model-Driven Architecture and Architecture-Driven Modernization worlds. They leverage cloud computing to provide cost-efficient and reliable services. However, unlike InCLOUDer, they focus only on the technical dimension of the migration but not on the organisational, policy- and business-related aspects as well as recommended by the literature [9] [10]. The authors of [11] presented a process framework to migrate legacy software to cloud environments based on their experiences in migrating a legacy open source system. They propose seven distinct activities for cloud migration including the cloud selection but they do not cope with application's constraints related to data privacy or other organisation's criteria.

Goal. This paper describes the process followed by InCLOUDer [12] to migrate applications to PaaS environments and the meta models that support the process. For the first time, this paper formalises and explains the meta models used by InCLOUDer and makes contribution on showing how they work when taking into account the explained six PaaS dimensions. More specifically, InCLOUDer uses meta models for application profiling, organisations PaaS criteria description, and PaaS offerings selection. Prior to migration, this paper addresses PaaS selection in respect of six PaaS identified dimensions [4]—the application, SLA, deployment, security, portability and usability dimensions—with emphasis in achieving portability to increase the level of service and minimise vendor lock-in. InCLOUDer assist the PaaS selection and supports the decision on how to adapt the application to the selected PaaS configuration.

2 Approach to Application Migration to PaaS

The InCLOUDer decision support system assists the migration of applications to PaaS environments. **Fig. 1** shows a high-level description of InCLOUDer and its three input models marked with red arrows. The organisation migrating the application to the PaaS specifies both the **organisation's criteria to PaaS migration**—which weights each of the six PaaS migration dimensions in relation to the importance of each one of them for the organisation; and the **application profile** capturing the application's requirements, constraints, and architecture according to the meta model in **Fig. 3**. The third input model lets PaaS vendors describe their PaaS offerings—or modellers if they do not—according to the meta model in **Fig. 2**.

Fig. 1 presents three of the mechanisms used in InCLOUDer, namely the **InCLOUDer application components re-scatterer**, the **InCLOUDer alternatives generator**, and the **InCLOUDer alternatives selector**. The **InCLOUDer application components re-scattering** and the **InCLOUDer alternatives generator** take into account the application profile and the organ-

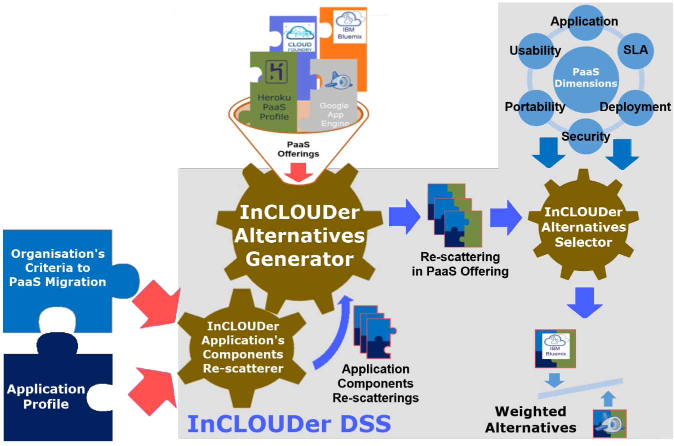
IBM Bluemix
CLOUD FOUNDRY
Heroku PaaS Profile
Google App Engine
PaaS Offerings
Application
Usability
SLA
PaaS Dimensions
Portability
Deployment
Security
InCLOUDer Alternatives Generator
Re-scattering in PaaS Offering
InCLOUDer Alternatives Selector
Organisation's Criteria to PaaS Migration
Application Profile
InCLOUDer Application's Components Re-scatterer
Application Components Re-scatterings
InCLOUDer DSS
Weighted Alternatives

Fig. 1. High-level description of the InCLOUDer decision support system (DSS)

isation's criteria to re-scatter the application's components to the local premises and the PaaS offerings. The reason to keep components in the local premises is to meetThe **InCLOUDer alternatives selector** weights each alternative to migrate the application—**re-scatterings in PaaS Offering**—to select the most beneficial alternative to migrate the application to PaaS—**weighted alternatives**.

Fig. 2 shows the meta model to describe and validate a PaaS offering. Each PaaSProvider supplies several PaaSOfferings which provides PaaSCharacteristics. PaaSCharacteristics include, exclude, or group other PaaSCharacteristics. The GroupRelationship can be an OR or XOR relationship depending on whether several PaaSCharacteristics can be present at the same time—OR—or only one of them—XOR. The PaaSCharacteristics present in a PaaSOffering determines the value of each Property of a PropertyType.

The meta model shown in **Fig. 3** lets organisations specify their application's requirements, constraints, and architecture. Organisations can model their application's data and application components, specify in which nodes these components run, and how these components are connected. Additionally, organisations can model the application's requirements and constraints.

Surajbali and Juan-Verdejo described the six PaaS migration dimensions with more detail in [4]. Once the organisation specifies the organisation's PaaS migration criteria and the two input models based on the meta models shown in **Fig. 2** and **Fig. 3**. InCLOUDer generates the alternatives to migrate to PaaS consisting in a model describing—according to the meta model presented in **Fig. 4**—how to re-scatter the application's components to the local premises and the PaaS offerings. This meta model specifies which components (*AbstractComponents*) to migrate to a PaaSProvider and which to keep in the LocalPremises. Aditionally, this meta model supports measuring the PaaSOffering score per DimensionType. Each DimensionType is one of the six criteria to PaaS migration.

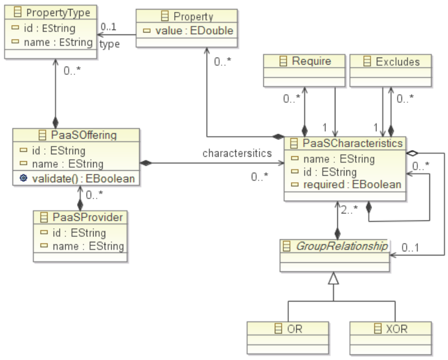

Fig. 2. PaaS Offerings Meta Model

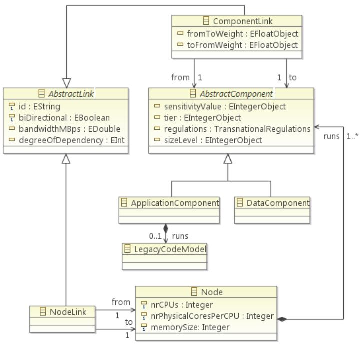

Fig. 3. Application Profile Meta Model

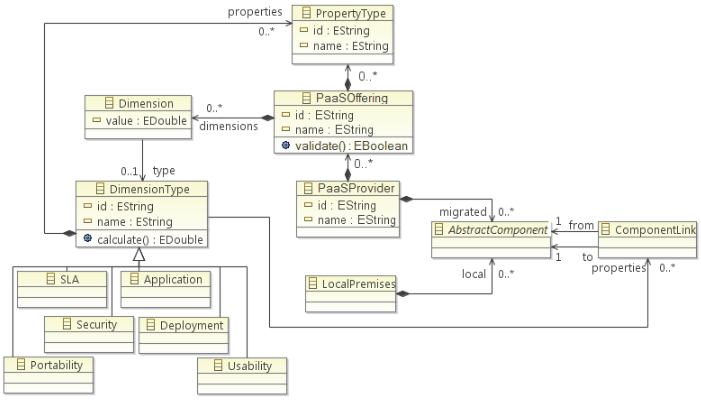

Fig. 4. Meta Model of the Application Re-scattered to PaaS Offerings

Contribution. The uniqueness of the InCLOUDer approach strives in re-scattering the application components to the local premises and the different PaaS offerings to maximise the organisation's PaaS migration criteria. All the proposed alternatives to migrate to PaaS comply with the application's constraints such as component-to-component communication delays or platform-dependent specifications—e.g. supplying a MySQL database management system or a Tomcat application server. Additionally, the PaaS migration alternatives respect the organisation's policies related to data protection and sensitivity by keeping locally components which contain sensitive data. The future work related to this work includes adding to the quantitative evaluation done for InCLOUDer the empirical validation of the approach taken by letting organisations use InCLOUDer for the migration of their applications to PaaS environments.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I.: A view of Cloud Computing. Communications of the ACM **53** (2010) 50–58
2. Juan-Verdejo, A., Surajbali, B., Baars, H., Kemper, H.G.: Moving business intelligence to cloud environments. In Proceedings IEEE INFOCOMM, Cross Cloud 2014, Toronto (2014)
3. Khajeh-Hosseini, A., Greenwood, D., Sommerville, I.: Cloud migration: A case study of migrating an enterprise it system to iaas. In: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, IEEE (2010) 450–457
4. Surajbali, B., Juan-Verdejo, A.: A marketplace broker for platform-as-a-service portability. In: Proceedings of the Third European Conference on Service-Oriented and Cloud Computing (ESOCC) (to appear), ACM (2014) 275–282
5. Joe, M.: Does platform as a service have interoperability issues (2010)
6. Gonidis, F., Simons, A.J., Paraskakis, I., Kourtesis, D.: Cloud application portability: an initial view. In: Proceedings of the 6th Balkan Conference in Informatics, ACM (2013) 275–282
7. Jamshidi, P., Ahmad, A., Pahl, C.: Cloud migration research: a systematic review. (2013)
8. Mohagheghi, P., Sæ ther, T.: Software Engineering Challenges for Migration to the Service Cloud Paradigm: Ongoing Work in the REMICS Project. In: 2011 IEEE World Congress on Services, Ieee (2011) 507–514
9. Juan-Verdejo, A., Baars, H.: Decision support for partially moving applications to the cloud: the example of business intelligence. In: Proc. of the int. workshop on Hot topics in cloud services, ACM (2013) 35–42
10. Ahmad, A., Babar, M.A.: A framework for architecture-driven migration of legacy systems to cloud-enabled software. In: Proceedings of the First International Conference on Dependable and Secure Cloud Computing Architecture - DASCCA '14, New York, New York, USA, ACM Press (2014) 1–8
11. Babar, M., Chauhan, M.: A tale of migration to Cloud Computing for sharing experiences and observations. In: SECloud '11 Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing. (2011)
12. Juan-Verdejo, A., Zschaler, S., Surajbali, B., Baars, H., Kemper, H.G.: InCLOUDer: A formalised decision support modelling approach to migrate applications to cloud environments. In: 40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2014), IEEE (2014)