Using Feature Modelling and Automations to Select among Cloud Solutions

Clément Quinton, Laurence Duchien, Patrick Heymans, Stéphane Mouton, Etienne Charlier

Using Feature Modelling and Automations to Select among Cloud Solutions

Clément Quinton, Laurence Duchien
Inria Lille - Nord Europe
LIFL UMR CNRS 8022
University Lille 1, France
{*clement.quinton, laurence.duchien*}*@inria.fr*

Patrick Heymans
PReCISE Research Center
University of Namur, Belgium
patrick.heymans@fundp.ac.be

Stéphane Mouton, Etienne Charlier
CETIC
Charleroi, Belgium
{*stephane.mouton, etienne.charlier*}*@cetic.be*

***Abstract*—Cloud computing is a major trend in distributed computing environments. Resources are accessed on demand by customers and are delivered as services by cloud providers in a pay-per-use model. Companies provide their applications as services and rely on cloud providers to provision, host and manage such applications on top of their infrastructure. However, the wide range of cloud solutions and the lack of knowledge in this domain is a real problem for companies when facing the cloud solution choice. In this paper, we propose to use *Software Product Line Engineering* (SPLE) and *Feature Model* (FM) configuration to develop a decision-supporting tool. Using such modelling techniques and automations, this tool takes into consideration the application technical requirements as well as the user quality requirements to provide an accurate result among cloud solutions that best fits both requirements.**

***Keywords*-Cloud Computing; Software Product Line Engineering; Feature Modelling; Separation of Concerns**

I. Introduction

Cloud computing has emerged as a major trend in distributed computing for "*enabling convenient, on–demand network access to a shared pool of configurable computing resources (*e.g.*, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*" [9]. In cloud computing, resources (processing, network and storage) are accessed on demand by customers and are delivered as services by cloud providers in a *pay-per-use* approach [2], [3]. This service provisioning model brings flexibility to companies that rely on cloud providers' infrastructure, *i.e.*, *Infrastructure-as-a-Service* (IaaS), to host their applications on virtual machines by configuring their whole software stack (operating system, libraries, applications servers). The deployment of such applications, called *Software-as-a-Service* (SaaS), can also be done on top of *Platform-as-a-Service* (PaaS) discharging companies from dealing with virtual machine configuration, by bringing an application support (typically an application server) and hiding the cloud infrastructure to make the deployment easier. When deploying an application into the cloud, companies have to deal with a wide range of resources at different levels of functionality among available cloud solutions. This leads to complex choices among cloud solutions which are usually made in an ad hoc manner. We argue that this selection can be systematized and partly automated using techniques originating from *Software Product Line Engineering* (SPLE) approach [4], [11], considering cloud providers as members of the same product family. Indeed, these providers offer a set of technical requirements (*e.g.*, an application server, a database) to host an application, some of them being shared (commonality) while others are different (variability) among providers. A well-known approach to variability modelling is by means of *Feature Models* (FMs) introduced as part of *Feature Oriented Domain Analysis* (FODA) [7]. By means of constraints between the application configuration and the cloud solutions features (*i.e.*, technical requirements), FM provides an interesting solution for selecting a cloud solution able to meet the application technical requirements. However, using FM configuration and constraint techniques is not efficient enough and the range of available cloud solutions corresponding to the application configuration is still significant. Among potential cloud solutions, a more accurate choice can be done based on specific customer quality requirements (*e.g.*, security, cost) or a combination of such requirements (or concern). Thus, crosscutting concerns for both application and cloud providers can be selected and prioritized according to the cloud solution offer. This paper presents the first results on this topic and describes a solution sketch for the problems that are usually met by companies.

The remainder of this paper is structured as follows. In Sec. II we briefly explain CETIC's business and how it is involved in cloud computing. We then describe in Sec. III how we think SPLE, and FMs in particular, can be used as the basis for the decision-support tool we plan to develop. Sec. IV discusses the future work.

II. The CETIC

A. Presentation

The *Centre d'Excellence en Technologies de l'Information et de la Communication* (CETIC) is an applied research center dedicated to *Information and Communication Technologies* (ICT) that works closely with several Belgian companies and universities and is involved in various European research projects. CETIC aims at supporting regional economic development, by helping companies in their ICT development. CETIC has expertise in three main domains,

software engineering, service-oriented technologies and embedded systems. The *Software and Services Technologies* (SST) department of CETIC brings an expertise in service-oriented architectures, distributed architectures and cloud computing. For the latter, initial work was focused on the domain of Grid Computing. Increasing demand from companies led CETIC to evolve towards management of the cloud computing infrastructure layer (IaaS). Currently, CETIC focuses its cloud computing research on PaaS environments and analyses how to develop and release software as SaaS. CETIC provides consulting services, feasibility studies, training and prototyping to companies that wish to deploy applications on top of PaaS or IaaS cloud providers.

B. CETIC's Cloud Business

Cloud computing has a major impact on SMEs which constitute the core of CETIC's customers. In particular, it allows companies to use virtual resources in a *pay-as-you-go* model. This elasticity of resources brings them reliability and flexibility when providing services to their customers, being able to scale on-demand (*e.g.*, scaling-up cloud resources for a given period to answer a huge load of requests). CETIC supports its customers (in most cases Belgian SMEs) by identifying the best candidate cloud solution that matches their requirements. Typically, CETIC's customers start new projects (*e.g.*, migration from a desktop application to cloud), start new businesses or want to replace their IT infrastructure. The application's architecture is studied to determine the best solution *i.e.*, the best PaaS or IaaS cloud provider (the former being most often chosen). The wide range of cloud providers to host the application makes the choice difficult, and there is a lack of visibility among them to select the best fit (*i.e.*, the one that matches best the application technical requirements). Without any decision-support tools, CETIC tends to recommend a cloud solution that has already been chosen for a previous customer to meet more or less similar expectations. Customers also give different priority levels to specific concerns to be addressed in cloud computing deployment such as security, scalability and cost. Therefore, we define two main challenges that CETIC (and all stakeholders involved in cloud deployment) have to face when looking for a cloud solution to host a customer's application:

- $\mathcal{C}_1$: *Identify candidate cloud solutions.* The essential point when deploying an application in the cloud is to ensure the compatibility between its architecture and functionalities, and the cloud provider's offer *before* the deployment to avoid a costly trial-and-error process.
- $\mathcal{C}_2$: *Select the most appropriate solution.* Among potential cloud solutions, a more accurate choice can be done, based on specific quality requirements (*e.g.*, security, cost) and priorities between such requirements.

III. FM as a Decision-Support Tool

When providing applications as SaaS, companies face a first choice as there are two different ways to achieve the deployment, *(i)* atop of PaaS or *(ii)* atop of IaaS environments. The former is the simplest way to get an application up and running in the cloud as it fully hides the infrastructure management, while the latter is more complex to setup but allows the configuration of the whole software stack used to run the application. We discuss both approaches and describe how FMs can address challenge $\mathcal{C}_1$.

A. Deployment on Top of PaaS

There are several PaaS cloud solutions one can choose to deploy an application, typically hosted on an application server (*e.g.*, Tomcat). When deploying an application on top of PaaS, one usually has to select a kind of application server to host the application, the development language, the kind of database (if needed), etc. When performed "manually", this selection is a tedious task. It could be entirely automated thanks to inter-feature dependencies (*i.e.*, constraints). Fig. 1 a) depicts such constraints. This example shows an excerpt of a FM of a simple application (**`FM_Application`**) written in **`Java`** and requiring a database, either **`MariaDB`**, **`PostgreSQL`** or **`MySQL`**. We also give an excerpt of a FM related to a given PaaS (**`FM_PaaS`**), chosen to deploy the application. Thanks to constraints between FMs, we ensure the application to be fully working once deployed, as every selected feature in the **`FM_Application`** is associated to a feature in the **`FM_PaaS`** (*i.e.*, the application configuration requirements are entirely covered by the PaaS technical funtionalities). Note that each existing PaaS cloud solution is associated to its related FM. Existing tools, such as FAMILIAR provide mechanisms to merge FMs together [1]. Merging the FMs of each PaaS solution yields a FM that gives a detailed view of all available PaaS solutions. This way one does not need to look for constraints between the application FM and every Paas FM. Using this approach, one possible outcome is that there is no PaaS available that fits the application technical requirements. In such a situation, the application can still be deployed as SaaS on top of IaaS provider, such as Amazon EC2[1].

B. Deployment on Top of IaaS

Deploying an application on top of a IaaS provider is slightly different than a PaaS where all infrastructure resources are provided transparently. The deployment on top of IaaS does not only require the application to be deployed, but also the configuration of the whole software stack (operating system, libraries, applications servers and applications), called *virtual appliance*. Such appliances run on virtual machines hosted by IaaS data centers. Fig. 1 b) depicts a deployment on top of IaaS. By means of constraints, FMs

[1] http://aws.amazon.com/ec2/

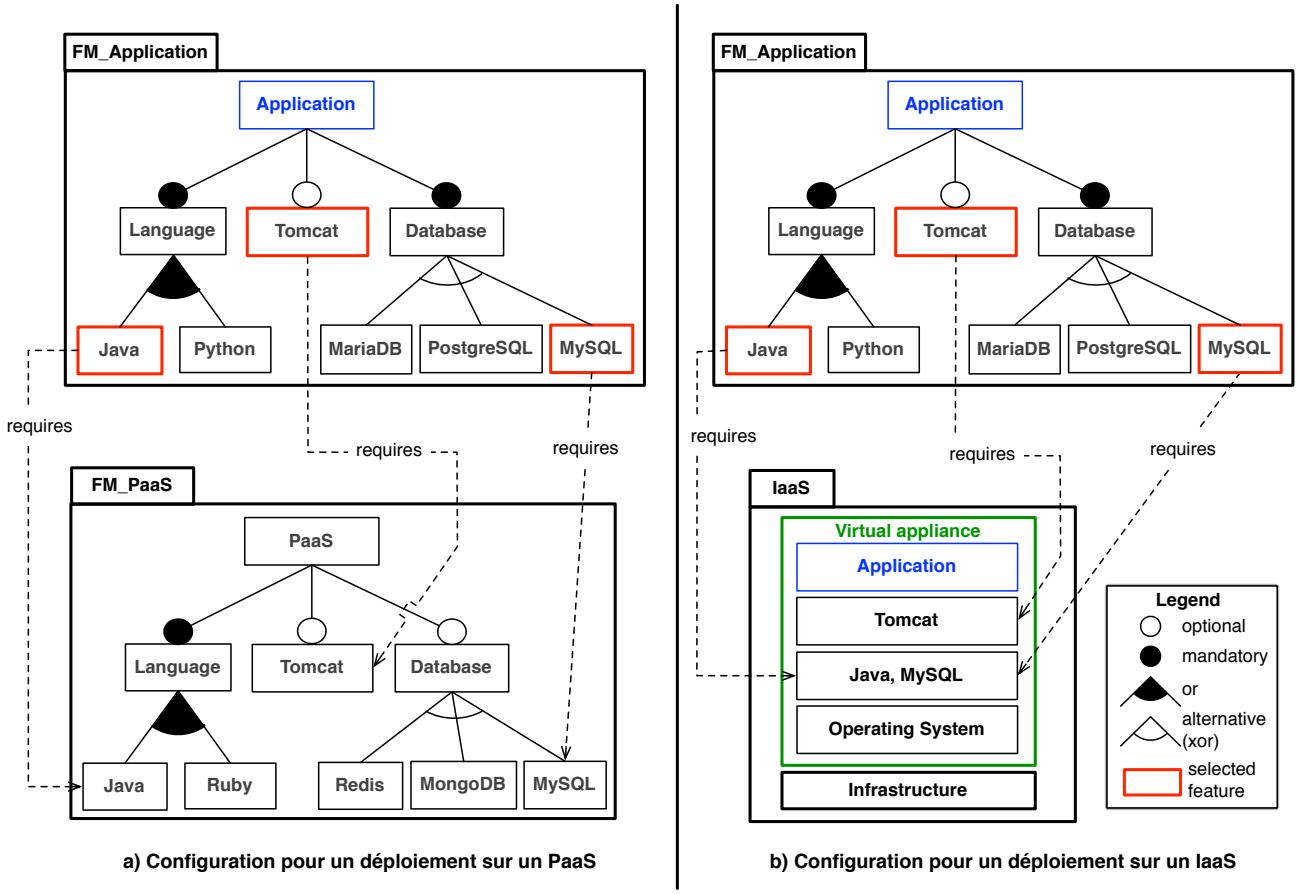

Figure 1. Different ways of configuring an application to be deployed in the cloud

are used to describe the whole virtual appliance needed to run the application. The application requires a **`Tomcat`** application server, a **`MySQL`** database and a **`Java`** library (typically a JDK). Configuring one's own PaaS this way allows the deployment of lightweight virtual appliances, as there is no unexpected tools added (*e.g.*, a version control system) like it often happens using public PaaS. If needed, these kinds of tools must be part of the **`FM_Application`** configuration. This point is very important in order to reduce the virtual appliance disk space footprint, as IaaS providers today face important problems of virtual appliance storage and slow transfer of virtual appliances data across cloud servers [12]. Reducing virtual appliance footprint also means reducing costs, as it is possible to run a given application on a smaller virtual machine instance (*e.g.*, on Amazon EC2, a large instance is four times more expensive than a small one for a Linux usage). In [6], they derive a set of virtual appliances using FM for an auto-scaling queue for a given application and IaaS while we use FMs to derive a virtual appliance for a given application and select the best IaaS provider corresponding to the virtual appliance configuration.

C. Considering User Quality Requirements: Challenges

Whether the application is deployed on top of PaaS or IaaS, the range of cloud solutions indicated by this decision-support tool is still significant. Based on specific customer requirements, we describe how the FM configuration can address challenge $\mathcal{C}_2$ and help such a tool to provide a more accurate result. The previously described approach works fine considering technical requirements (*e.g.*, if the application requires a **`Tomcat`** application server, the tool suggests PaaS solutions that provide such a functionality). However, it does not take into consideration the users' quality requirements. Customers often ask their application to be secure, flexible, reliable or inexpensive, and usually a combination of these criteria. Such criteria, or concerns, crosscut the cloud provider functionalities (*i.e.*, features), and managing FM related to several concerns is intuitively a problem of *Separation of Concerns* (SoC) [10]. To support SoC, we propose to use *extended feature models* [5], [8], where information (*e.g.*, quality requirements) can be attached to features. This information, defined at application level (*i.e.*, in the **`FM_Application`**) by the customer can be

compared to the different PaaS FM (*i.e.*, **`FM_PaaS`**) during the configuration phase thanks to inter-features constraints. Figure 2 shows an excerpt of a PaaS FM and its related concerns. Features can be related to one (**`SSL`**, **`Database`**) or several (**`Load balancer`**) concerns, and concerns can crosscut several features (**`cost`**).

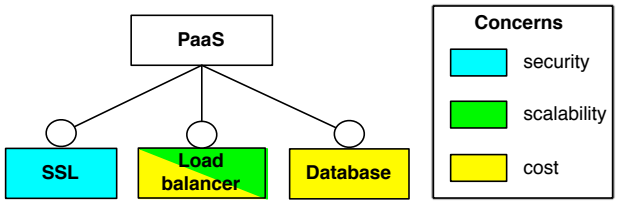

Figure 2. Separation of Concerns and Feature Model

Note that in a cloud environment, the **`cost`** concern (that customers often specify as the most important with **`scalability`**) is strongly related to resource usage (CPU, network and disk space). In the above example, the **`Load balancer`** feature, if selected, requires several application servers to be used and consequently more CPU consumption, while the **`Database`** feature requires more disk space. Attributes associated to features can specify a concern and a related value (*e.g.*, the **`Database`** feature can be associated to attributes like *size: 20GB, cost 15$* or a set of attributes).

IV. Discussion

In this paper, we have presented the problems that CETIC faces when trying to identify the best cloud solution for a given application. We first plan to develop a prototype decision-support tool that helps identifying candidates among all PaaS cloud providers. This tool, based on feature model automations, deals with variability in PaaS functionalities. Combined with SoC to take into account customer quality requirements and priorities, it provides an accurate range of PaaS solutions to host the application. Secondly, if there is no suitable PaaS solution (whether the technical requirements or the customer quality requirements cannot be met), the application can be deployed on top of IaaS. For such situations, we also consider developing a tool that generates all the virtual machine configuration script, in order to built a comprehensive solution that facilitates the choice of a cloud platform as well as the deployment step.

Many questions remain open, e.g., integrating results from the COTS selection literature, dealing appropriately with priorities, how to design the tool, and validating/improving the approach through case studies.

Acknowledgment

This work is supported by Ministry of Higher Education and Research, Nord–Pas de Calais Regional Council and FEDER through the *Contrat de Projet Etat Region Campus Intelligence Ambiante (CPER CIA) 2007-2013*.

References

[1] M. Acher, P. Collet, P. Lahire, and R. B. France. Composing feature models. In M. van den Brand, D. Gasevic, and J. Gray, editors, *SLE*, volume 5969 of *Lecture Notes in Computer Science*, pages 62–81. Springer, 2009.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. Technical report, 2009.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Future Gener. Comput. Syst.*, 25:599–616, June 2009.

[4] P. Clements and L. M. Northrop. *Software Product Lines: Practices and Patterns*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[5] K. Czarnecki, T. Bednasch, P. Unger, and U. W. Eisenecker. Generative programming for embedded software: An industrial experience report. In *Proceedings of the 1st Conference on Generative Programming and Component Engineering*, GPCE'02, pages 156–172, London, 2002. Springer-Verlag.

[6] B. Dougherty, J. White, and D. C. Schmidt. Model-Driven Auto-Scaling of Green Cloud Computing Infrastructure. *Future Gener. Comput. Syst.*, 28(2):371–378, Feb. 2012.

[7] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-Oriented Domain Analysis (FODA) - Feasibility Study. Technical report, The Software Engineering Institute, 1990.

[8] A. S. Karataş, H. Oğuztüzün, and A. Doğru. Mapping Extended Feature Models to Constraint Logic Programming over Finite Domains. In *Proceedings of the 14th International Conference on Software Product Lines*, SPLC'10, pages 286–299, Berlin, 2010. Springer-Verlag.

[9] P. Mell and T. Grance. The NIST Definition of Cloud Computing. Technical report, National Institute of Standards and Technology, 2009.

[10] D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, December 1972.

[11] K. Pohl, G. Böckle, and F. J. v. d. Linden. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[12] C. Quinton, R. Rouvoy, and L. Duchien. Leveraging Feature Models to Configure Virtual Appliances. In *Proceedings of the 2nd International Workshop on Cloud Computing Platforms (CloudCP)*. ACM, 2012. To appear.