Thesis Title

Christos Papoulas

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
Knossou Av., P.O. Box 2208, Heraklion, GR-71409, Greece

Thesis Advisors: Prof. *Kostas*, Dr. *Magoutis*

Optionally: This work was partially supported by **fill in sponsor name**

University of Crete
Computer Science Department

Your Title

Thesis submitted by
Author Name
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: ______________________________
Author Name

Committee approvals: ______________________________
Name of first member
Assistant Professor, Thesis Supervisor

____
Name of second member
Associate Professor, Committee Member

____
Name of third member
Professor, Committee Member

Departmental approval: ______________________________
Name of Director of Graduate Studies
Professor, Director of Graduate Studies

Heraklion, July 2015

Abstract

In this work ...

Περίληψη

Στην εργασία αυτή . . .

Acknowledgements

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Competitive pressures, globalization and technology expansion prompted the enterprises all over the world in the discovery of methods that will lead to an enhancement of their performance and their competitive standing. Therefore, the goal of the majority of firms is to apply new and innovative business models and to implement Information Technology solutions as quickly as possible; ideally, before their competition does [1, 2]. In alignment with the context we described above, IBM released an environment that allows developers to create, deploy and scale services and applications "in minutes rather months". The speed and the agility of the development of those applications deliver customer loyalty. This new approach that includes a pool of application services, development tools and languages renders businesses able to adjust fast enough in order to meet the needs of their customers by introducing new products to market faster. This means that developers must be equipped with tools, languages, interfaces, frameworks which deliver the desirable speed, such as DevOps innovations and composable cloud services [3]. The composition of software from components that are able to be plugged together is not a new idea. UNIX pipes allow the composition of simple linear programs where the first process writes to stdout and the subsequent process reads from stdin. As a result, an output stream is composed with an input stream. The AVS computer visualization systems give users the capability to create graphics applications by creating graphs that consist of simpler filter and rendering components [4]. Moreover, Common Component Architecture (CCA) is a system that puts together components in order to build large supercomputer simulations [5]. There are a lot more component systems and they differ in the behavior of the components and the method of composition [6]. "A component is a service that is defined by a set of interfaces that it implements. A user of a component invokes the functions defined by the services interfaces and the appropriate actions are taken by the component implementation. In most cases this service is considered to be stateless; however, in some cases the component interacts with a resource, such as a database which has state" [7]. Main advantage that emerges from component technology is the encapsulation of knowledge in small building blocks (components) that set aside concerns like how they are developed or where they reside and allow developers to focus on the design of the application itself and the integration of the components [8]. Chef and Puppet are two examples of configuration management systems that codify and execute procedures such as installation and deployment of cloud-based (and not only) applications around software components. They are based on model-driven design principles, Model Driven Engineering (MDE). Moreover, TOSCA [bla] and CloudML [bla] are two model-driven frameworks that describe application structure, requirements, and manage application deployments in a cross-platform manner [bla, bla]. Such frameworks are adopted by DevOps professionals and lead to the creation of communities that want to exchange know-how. That is how the need for a social networking platform specially tailored to the needs of DevOps professionals was created. This social networking platform (PaaSage SN) is created within the context of PaaSage project and follows a model-driven approach, which means that it allows the developers to represent and store models of applications and infrastructure and to use them in automating deployment on cloud [9]. The modeling language that is adopted by the PaaSage SN is CAMEL; it is actually a whole family of domain specific languages (DSL) that cover all the aspects of specification and execution of multi-cloud applications.

1.1 Motivation

1.2 Background

1.3 Methodology

1.4 Other section

1.5 Related Work

Chapter 2

Management of ...

General discussion . . .

2.1 AA

2.2 BB

2.3 CC

2.4 DD

Chapter 3

Methodology

General discussion . . .

3.1 AA

3.2 BB

3.3 CC

3.4 DD

3.5 EE

3.6 FF

Chapter 4

Evaluation

General discussion ...

4.1 AA

4.2 BB

4.3 CC

4.4 DD

4.5 EE

4.6 FF

Chapter 5

Comparison

Compare your work . . .

5.1 AA

5.2 BB

5.3 CC

5.4 DD

5.5 EE

5.6 FF

Chapter 6

Conclusions and Future Work

Bibliography

[1] H. Chesbrough and R. S. Rosenbloom, "The role of the business model in capturing value from innovation: evidence from xerox corporation's technology spin-off companies," *Industrial and Corporate Change*, vol. 11, pp. 529–555, 2002.

[2] R. L. Kumar, "A framework for assessing the business value of information technology infrastructures," *J. Manage. Inf. Syst.*, vol. 21, no. 2, pp. 11–32, Oct. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1277694.1277696

[3] W. Falk, C. Ferris, M. Fork, and R. Sauerwalt, "Accelerate development of new enterprise solutions for the cloud with codename bluemix," IBM Redbooks, 11400 Burnet Road, Austin, TX 78758-3493, USA, Tech. Rep., 2002. [Online]. Available: http://www.redbooks.ibm.com/redpapers/pdfs/redp5011.pdf

[4] I. Curington and M. Coutant, "Avs: A flexible interactive distributed environment for scientific visualization applications," in *Proceedings of 2nd Eurographics Workshop on Scientific Visualization*, 1991.

[5] R. Armstrong, D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, and B. Smolinski, "Toward a common component architecture for high-performance scientific computing," in *Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing*, ser. HPDC '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 13–. [Online]. Available: http://dl.acm.org/citation.cfm?id=822084.823232

[6] D. Gannon, S. Krishnan, A. Slominski, G. Kandaswamy, and L. Fang, "Building Applications from a Web Service based Component Architecture," in *ICS 2004 Workshop on Component Models and Systems for Grid Applications.* Kluwer Academic, 2004. [Online]. Available: http://www.extreme.indiana.edu/xcat/publications/stmalo-gannon.pdf

[7] D. Gannon, S. Krishnan, L. Fang, G. Kandaswamy, Y. Simmhan, and A. Slominski, "On Building Parallel and Grid Applications: Component Technology and Distributed Services," in *CLADE 2004, Challenges of Large Applications in Distributed Environments.* IEEE Computer Society Press, 2004. [Online]. Available: http://www.extreme.indiana.edu/xcat/publications/clade04-cca-grid.pdf

[8] A. D. Malony, S. Shende, N. Trebon, J. Ray, R. C. Armstrong, C. E. Rasmussen, and M. J. Sottile, "Performance technology for parallel and distributed component software," *Concurrency - Practice and Experience*, vol. 17, no. 2-4, pp. 117–141, 2005. [Online]. Available: http://dx.doi.org/10.1002/cpe.931