Ontology-driven Framework for Self-management in Cloud Application Platforms

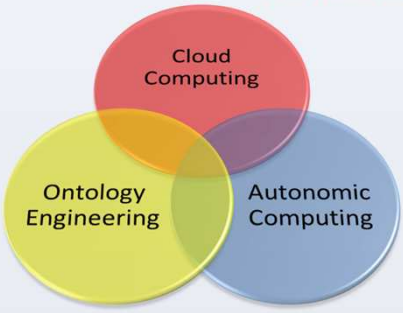
Rustem Dautov

South East European Research Centre (SEERC)

International Faculty , the University of Sheffield

INTRODUCTION

- Cloud application platforms can perform simple adaptation actions only at the IaaS level (i.e. load balancing and elasticity)
- Existing cloud application platforms are unable to perform more **complex and intelligent adaptation scenarios** (e.g. modifying the actual structure and/or behavior of an application during its runtime)
- The goal of this work is to achieve a self-managing cloud application platform by **utilising ontologies and rules to represent self-reflective knowledge and adaptation policies**



KEY RESEARCH CHALLENGES

- **Enabling "near real-time" adaptations** (i.e. no delay between the moment when a problem is detected and the moment when an adaptation action takes place)
  - **Identifying parameters** to be monitored and parameters that can be neglected (so called "noise")
  - **Minimising the impact** of the framework's presence in the system when deployed to a cloud
  - **Predicting all possible effects** of adaptation actions and avoiding potential conflicts

FRAMEWORK OVERVIEW

4. In order to process constantly flowing streams of monitored data coming from sensors, we will be using techniques from:
   - **Complex Event Processing (CEP)**
   - **Stream Reasoning**

5. Using existing reasoning engines will exempt us from implementing an analysis engine from scratch

6. Based on the results obtained at the analysis stage, the framework will produce an adaptation plan – a set of actions to be executed in order to cope with a problem

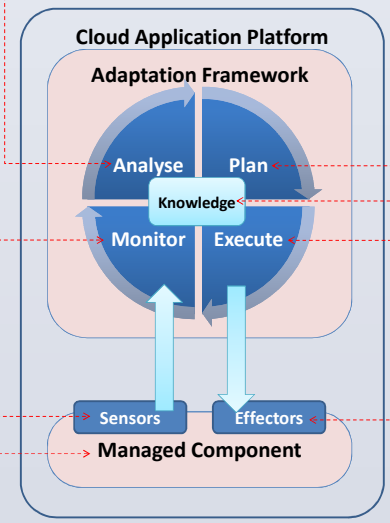
3. **Sensors** are software components for collecting information about a managed component

7. An adaptation plan is executed at this stage by means of effectors

2. A **managed component** could be either a deployed application or the application platform itself

8. Ontologies and rules are to be used in order to:
   - **represent self-reflective knowledge** (e.g. internal structure of the system, relations between subcomponents, available resources, etc.)
   - **define adaptation policies**

1. The framework builds upon IBM's **MAPE-K** reference model. According to this model, an adaptation loop includes 4 steps: **M**onitoring, **A**nalysing, **P**lanning, and **E**xecution. All these activities are performed based on a shared **K**nowledge base.



9. **Effectors** are software components for executing adaptation instructions

WHY ONTOLOGIES?

- With reasoning capabilities which come along with this approach there is no need to implement an analysis engine from scratch
- Adaptation policies are separated from programming code (i.e. separation of concerns)
- Adaptation policies can be developed by domain specialists, not professional programmers
- More flexible and intelligent adaptations at any level of abstraction
- More generic approach: also applicable to the IaaS level, as well as to any other distributed system
- Increase in reuse, automation and reliability

ACKNOWLEDGEMENTS AND CONTACTS

- This research work is funded by the Marie Curie Initial Training Network RELATE (www.relate-itn.com)

RELATE FP7 Marie Curie ITN
Trans-European Research Training Network on Engineering and Provisioning of Service-Based Cloud Applications

- Rustem Dautov is currently a PhD Candidate and Research Associate in the Information & Knowledge Management Research Group at SEERC.

rdautov@seerc.org Tel.: +30 2310 253477 Mob.: +30 6945 069303