Autonomic computing system for self-management of Machine-to-Machine networks

Mahdi BEN ALAYA, Salma MATOUSSI, Thierry MONTEIL, Khalil DRIRA
{ben.alaya, smatouss, monteil, khalil}@laas.fr
1. Introduction
2. Problematic
3. Related works
   - M2M communications standards
   - Autonomic computing
4. M2M Autonomic system principles
   - Autonomic manager architecture
   - Autonomic managers plug & play
   - Autonomic system for M2M networks
5. Experiments with ADREAM smart building
6. Conclusion and perspectives
Introduction

- Emergence of fixed and mobile communicating objects.
- Decrease of the communication cost.
- Improvement of networks performance.
- Availability of advanced dedicated services platforms.

Rise of new highly distributed ambient environment called Machine to Machine (M2M).

M2M is the ability for machines to communicate in real time without human intervention to streamline processes, optimize resources management and reinvent workflows and business processes.
M2M networks contains:
- Heterogeneous communicating machines.
- Different communication pattern type are required.
- Variety of applications deployed by various providers.
- Different vertical domains.
- Highly dynamic environment.

- Problematic: Increasing complexity of M2M communications: Increasing the cost of development, maintenance and research in M2M.

- Challenge: Design a generic autonomic M2M system based on M2M standards, autonomic computing paradigm, and decision models for self-management of communication, services and applications within M2M networks.
- ETSI published a set of standards for M2M:
  - M2M services requirements
  - M2M interfaces
  - M2M functional architecture

- ETSI did not addressed the self-management of M2M communication, service and application issues.

- ETSI started new specifications draft on usage of semantic in M2M.
  - Study on Semantic support for M2M
IBM proposed in 2001 the Autonomic Computing reference architecture.


Recent works:
- **SAF**: Symptom Automation Framework is a catalog based XML collaborative knowledge framework that enables diagnosis and treatment of complex system.
- **GAMF**: Generic Autonomic Management Framework aims to enable developing autonomic manager on any specific target system without re-implementing.
- **GRYPHON**: an IBM project focusing on the design and development of highly scalable, available and secure publish/subscribe systems.
- **FACUS**: Framework for Adaptive Collaborative Ubiquitous Systems. It proposes a Generic Collaboration Ontology and aims to deploy collaborative session for users based on event.
Autonomic manager architecture

- Managed Entities
- Sensors
- Effectors

- Knowledge Base
  - Analyzer
    - Inference Engine
  - Planner
    - Inference Engine
  - Executor
    - Inference Engine

- Monitor
  - Symptoms
  - Inference Engine

- Events
  - Change requests

- Actions
  - Action Plans
  - Events
Autonomic managers are the basic building blocks of autonomic systems.

They can be considered as services within service oriented architecture and might simultaneously be providing and consuming services.

Autonomic computing system
Autonomic managers should be adapted for resource-constrained devices and provide standardized mechanisms for seamless integration of these autonomic managers.

By defining:

- Transport-neutral mechanisms to address autonomic managers hosted services
- Discovery protocol to locate available services
- Generic protocol for accessing service-based resources representation
- Messaging protocol that allows services to subscribe to event notifications
- Security mechanisms to define how integrity and confidentiality can be ensured.
Autonomic managers are implemented based on the Device Profile for Web Services specification using the Java Multi Edition DPWS Stack (JMEDS).

DWPS is based on existing WS specifications such as
- WS-Addressing
- WS-Discovery
- WS-Eventing
- WSMetadataExchange
- WS-Transfer
- WS-Security
M2M Autonomic Computing system overview

M2M Autonomic system principles
Experiments in the SM domain
Conclusion and perspectives

Introduction
Problematic
Related works

M2M Network

M2M Gateway
M2M Server
M2M Device

M2M Autonomic computing system system overview
Experimentation

Introduction

Problematic

Related works

M2M autonomic system principles

Experiments in the SM domain

Conclusion and perspectives

M2M smart metering use case
Conclusion and perspective

• Autonomic computing system based on ETSI M2M architecture. And composed of a set of interoperable autonomic managers to self-manage ubiquitous complex situation.

As future work we propose to:
• calculate the overload that the framework generate.
• Experiment our solution in more complex scenario to self-manage a high number of heterogeneous machines.
• Introduce some artificial intelligence algorithms and mechanism in autonomic managers such as machine learning, negotiation and multi-agent systems.
• Contribute on M2M standardization within ETSI for the autonomic computing and semantic aspects.
Thank you for your attention