Towards Self-Managed Web of Everything
by Vagan Terziyan
Presenter’s short BIO and “self-(re)configuration” example

- Professor in Software Engineering;
- Head of Artificial Intelligence Dep.;
- Head of “MetaIntelligence” Research Lab.
- … in Kharkov National University of Radioelectronics (Ukraine)

“Self-reconfigurable“ means that the system is capable of utilizing its own system of control to change its overall structural shape.

- Professor in Distributed Systems;
- Head of Industrial Ontologies Group …
- … in the Department of Mathematical Information Technology, University of Jyvaskyla (Finland)
Researchers

- **Vagan Terziyan** (Head)
- Olena Kaykova
- Oleksiy Khriyenko
- Sergiy Nikitin
- Michal Nagy

- Michael Cochez
- Joonas Kesäniemi
- Viljo Pilli-Sihvola
- Jose Luis Garduno

Contact Person:

**Timo Tiihonen**

- **e-mails:**
  - timo.tiihonen@jyu.fi
  - vagan.terziyan@jyu.fi

- **phone:** +358 14 260 2741

**URL:** [http://www.mit.jyu.fi/ai/OntoGroup](http://www.mit.jyu.fi/ai/OntoGroup)
Agent Technology
Semantic Technology
SOA and Cloud Computing
Ubiquitous Computing
Social Computing
Artificial Intelligence

Self-Configurable Systems: Is it Hollywood Invented?
Why self (re)configuration?

**Versatility:** Self-reconfigurable systems are potentially *more adaptive* than conventional systems. The ability to reconfigure allows a system to *disassemble and reassemble components* to form new morphologies that are better suited for new tasks.

**Robustness:** Since system components are interchangeable (within a system and between different systems), self-reconfigurable systems can also *replace faulty parts autonomously*, leading to *self-repair*.

**Low Cost:** Self-reconfigurable systems can potentially lower overall cost by making many copies of one (or relatively few) type of modules so economies of scale and mass production come into play. Also, a range of a system can be made from one set of modules, saving costs through *reuse and generality* of the system.

One source of inspiration for the development of these systems comes from *biological systems that self-construct out of a relatively small repertoire of lower-level building blocks* (cells or amino acids, depending on the scale of interest). This architecture underlies the ability of biological systems to *physically adapt, grow, heal, and even self-replicate* - capabilities that would be desirable in many engineered systems.
Dynamic reconfigurability is reconfigurability at run-time, but not necessarily based on system self-awareness and intentions and not necessarily supported by any special algorithms.

Self-configurability is going one step beyond – the system is expected to autonomously and deliberately perform the reconfiguration.
What is configuration?

- **Composition** (content: components of data and capabilities),
- **Structure** (partonomy, business logic and interaction applied to content)
- and **Parameters** (features of the content and structure)
What is configuration of a self-configurable system?

Environment ("Policies")

System

Engine ("Soul")
Utility function ("Mind")
Composition and Structure ("Body")
Data and Knowledge ("Beliefs")
Capabilities ("Skills")

Parameters ("Features")

Everything is configurable!
Wanted: self-configuration within Internet

FP7: ICT Work Program for 2011-2012
Objective ICT-2011.1.3 Internet-connected objects

a) An open networked architecture for Internet-connected objects... The architecture should maximise interoperability ..., allow for re-use of object entities in the physical world across several application domains, and provide a coherent framework with open interfaces to manage the physical entities. Due to the mobility of objects and multiplicity of applications contexts, the architecture should support self-management, self-configuration and self-healing properties as well as scalable look up and discovery of "Internet of Things" resources and services and their subsequent mapping onto entities of the real world.

b) Adaptive software supporting data acquisition from a large number of sensors and providing integration with mainstream business platforms and components. Focus is on software to interpret the environmental and context information, detect information related to human intentions/behaviours, enable human-like inferences and multi-modal interactions, and eventually act on behalf of the users’ intentions...
Web 1.0 connects people to the content of static pages published in the World Wide Web.

In Web 1.0, a small number of writers create Web pages for a large number of readers.

Number of Public Web Pages

1990: 1
1998: 26 million (~26,000,000)
2008: >1 trillion (~1,000,000,000,000)

... more than the number of neurons in the human brain

Are we happy with it? Is it good for real technical applications?
Web of Humans (Social Web, Web 2.0)

Human Communities

Web 2.0
- Wikis
- Blogs
- Mashups
- Social Networks

Facilitates Human-to-Human interaction

HaaS: Human-as-a-Service
HaaU: Human-as-a-User
Web of Things

Machines, devices, computers, etc.

Facilitates Machine-to-Machine interaction

Ubiquitous Computing
Barcodes and RFID
Embedded Systems
Sensor Networks
Smart Spaces

DaaS: Device-as-a-Service
DaaU: Device-as-a-User
Facilitates Software-to-Software interaction
Pragmatic View to the Environment

What for and how can I use it?
Everything as a Capability

"I am capable of ..."

"... ironing"

"... cooking"

"... loving"

"... taking care"

"... repairing"
Capability: Product vs. Service
**SaaS** is a software application distribution and delivery model in which an application is hosted by a vendor or a service provider and its capabilities are made available for the use to a requester over the Internet.

**Advantages**
- Accessible from anywhere with an internet connection;
- No local server installation;
- Pay per use or subscription based payment methods;
- Rapid scalability;
- System maintenance (backup, updates, security, etc) often included in service;
- Possible security improvements, although users with high security requirements (e.g., large corporations) may find SaaS a security concern;
- Reliability.
What is Service-Oriented Architecture?

• **SOA** is the practice of sequestering the core business functions into independent services that don’t change frequently. SOA is a tool for software (as a service) integration. Rather than defining an API, SOA defines the interface to remote Web-based services in terms of protocols and functionality.

**Service Oriented Architecture (SOA)** is a means of designing and building software. It is a manufacturing model.

**Software as a Service (SaaS)** is a means of receiving software through an external party to your business similar to telephone or power utilities. It is a sales and distribution model.

[J Natoli, Intel]
Cloud Computing: Outsourcing Infrastructure

“Taxi Business”
Cloud

User application
Knowledge and Data Collections

Facilitates Knowledge-to-Knowledge interaction

Web of Knowledge (Semantic Web, Web 3.0)

Web of Knowledge
Ontologies
Metadata
Reasoning
Semantic Technology
Integration

KaaS: Knowledge-as-a-Service
KaaU: Knowledge-as-a-User
Most of the Current Web (dumb links)
Semantic Web
(data connected by relationships)
Linked Data:
“Bridges” between “Islands” of Data
Summary: What is Semantic Web?

- The **Semantic Web** is an evolving development of the World Wide Web in which the meaning (semantics) of information and services published on the Web and their inter-relationships are explicitly defined, making it possible for the Web-based software tools, agents, applications and systems to discover, extract and *understand* Web information resources and capabilities and automatically utilize it.

- **Semantic Technologies** are designed to standardize and support interoperability and integration of information content and capabilities (services) of Web-based systems and components at local and global scale.

- As a **software technology**, semantic technology encodes meanings separately from data and from application code to enable machines to understand, share and reason with them at execution time.
Why Semantic Web? (Ora Lassila)

A DIFFICULT MESSAGE

• Any specific problem (typically) has a specific solution that does not require Semantic Web technologies

• Q: Why then is the Semantic Web so attractive?
  A: For future-proofing

Semantic Web can be a solution to those problems and situations that we are yet to define

(seriously, I am not kidding...)

• “Semantic Web is about to reach its full potential and it would be too costly for companies not to invest to it…”

(Ora Lassila, Nokia Research Center (Boston), IASW-2005, Jyvaskyla)
Web of Intelligence (Distributed AI, Web 4.0)

Intelligent Agents and Applications

Web of intelligent entities (intelligence services), browseable, searchable, composable, self-managed, dynamic, mobile ...

Facilitates Intelligence-to-Intelligence interaction

Web of Intelligence

Agents and MAS

Data and Web Mining

Machine Learning

Self-Management

Context-Awareness

IN TaaS: Intelligence-as-a-Service

IN TaaS: Intelligence-as-a-User
What is an Agent?

ENVIRONMENT

Events

Behavior
Intelligent Agents

Software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing employ some knowledge or representation of a user’s goals or desires.

*IBM, Intelligent Agent Definition*
What is an Intelligent Agent?
Self-Configurability!

Events
Self-Configuration
Behavior
ENVIRONMENT
Internal Environment:
architecture, goals, capabilities, sensors, effectors, profile, knowledge, beliefs, etc.

External Environment:
user, other humans, other agents, applications, information sources, their relationships, platforms, servers, networks, etc.

Balance

Internal Environment:
arquitectura, objetivos, habilidades, sensores, efectores, perfil, conocimiento, creencias, etc.
Intelligent Agent is an entity that is able to *keep continuously balance between its internal and external environments* in such a way that in the case of unbalance agent can:

- *change external environment* to be in balance with the internal one … OR
- *change internal environment* to be in balance with the external one … OR
- find out and *move to another place* within the external environment where balance occurs without any changes … OR
- closely *communicate* with one or more other agents (human or artificial) to be able *to create a community*, which internal environment will be able to be in balance with the external one … OR
- *configure sensors* by filtering the set of acquired features from the external environment to achieve balance between the internal environment and the deliberately distorted pattern of the external one. I.e. “if you are not able either to change the environment or adapt yourself to it, then just try not to notice things, which make you unhappy”
Agent \(_i\) alone may “play” a function:
\[
\varphi_i = f^i (\omega_i),
\]
Where \(f^i\) is individual behavior.

**Agent and the World**

\(W = \text{Internal} + \text{External Environments}\)

- "Behavior" function
- "Read Only" world
- "Read and Write" world
- "Write Only" world

\(\omega_i\) and \(\varphi_i\) represent different types of interaction with the environment.
Indirect Collaboration (via communication)

Agent \(_i\) and Agent \(_k\) together may “play” 6 types of functions:

\[
\varphi_i = f^i (\omega_i); \quad \varphi_k = f^k (\omega_k);
\]

\[
\varphi_i = f^{ki} (\omega_k); \quad \varphi_k = f^{ik} (\omega_i);
\]

\[
\varphi_i = F^1 (f^i (\omega_i), f^{ki} (\omega_k)); \quad \varphi_k = F^2 (f^k (\omega_k), f^{ik} (\omega_i)),
\]

where \(F\) - collaborative behavior
Indirect Control (via environment)

Agent \(i\) may “play” 2 types of functions knowing rules of the environment:

\[ \varphi_i = f(\omega_i); \varphi_j = R(f(\omega_i)), \]

where \(R\) – function of environmental rules
Rented Service vs. Proactive Service
Proactive Web-Services: adding an agent to service platform — allows agent-based S2S communication
Summary: Why Agents?

- Growing complexity of computer systems and networks
- Distributed nature of systems (data, software, users, etc.)
- Ubiquitous computing, “Internet of Things” scalability challenges
- Need for self-manageability of a complex system
- Need for new software development paradigms in designing distributed systems
- Agent-based approach meets the above challenges …

And finally: Agents are excellent tool for self-configuration !!!
Context views, weights, masks and filters

Facilitates Context-to-Context interaction

Web of Context

Context as a viewpoint
Context-based filtering
Context-driven search
Context-driven ranking
Quality of Resource
Context-discovery

CONTaaS: Context-as-a-Service
CONTaaU: Context-as-a-User
Formal relationships, constraints, limitations, laws, mathematical models, defined business logic, communication protocols, goals/tasks definitions, permissions, prohibitions, commitments, conventions and other policies.

Facilitates Policy-to-Policy interaction.

Web of Policies
- Policy-Based Control
- Policy-Based Reasoning
- Policy-Based Coordination
- Policy negotiations
- Policy integration

PaaS: Policy-as-a-Service
PaaU: Policy-as-a-User
Multiple Policies

- Each industrial resource can theoretically be involved in several processes (organizations, relationships), and appropriate commitments (policies) of each process are applied to it, which can be either supplementary or contradictory. This means that the resource is part of several more complex resources and its role within each of the resource might be different. Modeling such resources can be provided by appropriate resource agent, which can make clones of itself and distribute all necessary roles among them.
… is the Web of “partonomy” (a classification based on part-of relation; not the same as taxonomy, which is a classification based on similarities). Configuration of an object (parts and their relationships) together with all policies applied to these parts fully describes the object from inside.
Proactive Configuration

Part_of product hierarchy in the ontology results to hierarchical MAS
Web of Presentations

... is the Web of visualization providers (or “metaproviders” according to 4i (“for-eye”) technology). The same content (either static or dynamic, homogeneous or heterogeneous) will be presented (and if needed also filtered and mashed) by different ways by different visualization providers.

Facilitates Presentation-to-Presentation interaction

Web of Presentations

Semantic Mash-Ups
Context-based presentation
Presentations Web Browser
Cloud of Visualizators
Proactive Visualization

VISaaS: Visualization-as-a-Service
VISaaU: Visualization-as-a-User
4i Philosophy: Visualization-as-a-Service

Web of Contexts

Web of Visualization Service Providers

Web of Configurations

Web of Policies

Web of Things

4iBrowser

FOREYE TECHNOLOGY
According to these visions of future Web, interoperability and collaboration will be possible only within mentioned groups of resources.

However future Web applications and Web-based systems will contain heterogeneous components and therefore will demand support for integration, interoperability, collaboration and mutual service provisioning between resources of different types.
Components of a modern system are not only highly heterogeneous but also globally distributed (SOA) ...
... or some processes and components may be placed and run within huge remote data centers (Cloud Computing)...

IaaS: Infrastructure-as-a-Service
... and some parts of the system may be placed into mobile terminals under supervision of various mobile ecosystems...
... also various systems should enable integrating them to a more complex business logic with other systems.
...and there should be an easy way to design, use, administrate and reconfigure the system.
A system should be open and ready to reconfigure itself when needed (1)
A system should be open and ready to reconfigure itself when needed (2)
A system should be open and ready to reconfigure itself when needed (3)
Even a business logic of a system can be imported and reconfigured on-the-fly (1)
Even a business logic of a system can be imported and reconfigured on-the-fly (2)
Even a business logic of a system can be imported and reconfigured on-the-fly (3)
Adding a "virtual representative" to every resource solves the global interoperability problem. Intelligent agent (a kind of “software robot”) will act, communicate and collaborate on behalf of each Web resource.
GUN – Global Understanding Environment

GUN = Global Environment + Global Understanding = Proactive Self-Managed Semantic Web of Everything

http://www.mit.jyu.fi/ai/OntoGroup/projects.htm
GUN can be considered as a kind of Ubiquitous Eco-System for Ubiquitous Society, which will be such proactive, self-managed evolutionary Semantic Web of Things, People and Abstractions where all kinds of entities can understand, interact, serve, develop and learn from each other.
Ψ-Projection of GUN-Related Research

**PΣI - projection:**

- **Proactivity** (agent technologies, Distributed AI, MAS, …)
- **Semantics** (Semantic Web, Semantic Technologies, …)
- **Services** (SaaS, SOA, SWS, Cloud Computing, …)
- **Intelligence** (self-management, machine learning, data mining, knowledge discovery, pattern recognition, NLP, …)
Our Roadmap: GUN-GERI-UBIWARE-SmartResource

GUN (Global Understanding Environment) – Proactive Self-Managed Semantic Web of Things – Web 5.0 candidate - general ecosystem and final destination

GERI (Global Enterprise Resource Integration) – GUN subset related to industrial domains - is based on PRIME (Proactive Inter-Middleware) as UBIWARE extension

UBIWARE – middleware for GERI

SmartResource – semantic technology, pilot tools and standards for UBIWARE
SmartResource project - our first step to GUN


One of the most essential results of the SmartResource project was creation of the “Smart Resource Technology” for designing complex software systems. The technology allows considering each traditional system component as a “smart resource”, i.e. proactive, agent-driven, self-managing. Such approach has shown certain advantages comparably to other software technologies, e.g. OOSE, SOA, Component-Based SE, Agent-Driven SE, Semantic SE, etc.
Smart Maintenance Environment

“Devices with on-line data”

“Experts”

“Services”

Maintenance data exchange

On-line learning

Maintenance data exchange
Challenge 1: General Adaptation Framework

Universal reusable semantically-configurable adapters

Semantic Agent Programming Language (RDF-based)

Universal reusable semantically-configurable adapters
Challenge 2: General Proactivity Framework

Role
"Feeder"
description

Role
"SCADA"
description

Role
"Maintenance worker"
description

Universal reusable semantically-configurable behaviors
Challenge 3: General Networking Framework

Scenario
"Predictive maintenance" description

Scenario
"Data integration" description

Universal reusable semantically-configurable scenarios for business processes
Resource Maintenance Lifecycle and Semantic History Collection

States
- Condition Monitoring
- Measurement
- Data Warehousing
- Industrial Resource
- Predictive Maintenance
- Fault detection, alarms

Symptoms
- Symptoms
- Predictive Measurement
- Predictive Monitoring
- Conditions Warehousing
- Diagnoses Warehousing
- Diagnoses

Diagnostics
- Maintenance Plan
- Plan Warehousing
- Fault identification, localization

Maintenance Plan
- Maintenance Planning
Due to heterogeneity of provided services and supported components, UBIWARE is based on integration of several technologies: Semantic Web, Distributed Artificial Intelligence and Agent Technologies, Ubiquitous Computing, SOA (Service-Oriented Architecture), Web X.0, P2P and related concepts.

The research and design on UBIWARE is started by Industrial Ontologies Group within UBIWARE project: “Smart Semantic Middleware for Ubiquitous Computing” (June 2007 – December 2010) funded by Tekes and industrial companies.

Project web page:

S-APL – is a hybrid of semantics (metadata / ontologies/ rules) specification languages, semantic reasoners, and agent programming languages. It integrates the semantic description of domain resources with the semantic prescription of the agents' behaviors.

http://users.jyu.fi/~akataso/sapl.html
Latest Innovations Invented by Industrial Ontologies Group in UBIWARE

- OntoNuts
  - OntoNuts – is the ontology-based instrument to reconfigure and enhance complex distributed systems by automated discovery and linking external sources of heterogeneous and dynamic data and capabilities during system runtime.

- 4i (“for eye”) technology
  - 4i – is smart ontology-based visualization technology able to automatically discover and utilize external visualization service providers and dynamically create and visualize mashups from external data sources in a context-driven way.

- Smart Comments
  - Smart Comments – is smart ontology-based technology for end-user-driven control and configuration management of the application in runtime based on smart mapping of appropriate tags from natural language comments provided by a SW engineer and the source code.
New semantics of RDF Statement in S-APL (object - executable resource)

Semantics of such statement means that the value of the \texttt{Property}_m of the \texttt{Resource}_i can be obtained as a result of execution of the procedure (query, service, function, etc.) represented as \texttt{Resource}_j.

S-APL

Semantic Agent Programming Language (Designed by Industrial Ontologies Group)
John :isInLoveWith :Mary.
“Executable Knowledge”
example 2

```
exe:Q2 rdf:type exe:SQL_Query.
exe:Q2 exe:hasSQL_SELECT_Type “AVG”.
exe:Q2 exe:hasSQL_SELECT_What “JournalPapers”.
exe:Q2 exe:hasSQL_SELECT_To “hasAvgYoungDoctStudentsPerformance”.
exe:Q2 exe:hasSQL_FROM “AI_Department”.
exe:Q2 exe:hasSQL_WHERE “Title = ‘PhD_Student’ AND Age < ‘30’ “.
exe:Q2
```

```
:AI_Department :hasAvgYoungDoctStudentsPerformance exe:Q2.
```

```
SELECT AVG(journalPapers) AS hasAvgYoungDoctStudentsPerformance FROM AI_Department WHERE Title = ‘PhD_Student’ AND Age < ‘30’
```

```
:AI_Department :hasAvgYoungDoctStudentsPerformance ‘7’.
```

```
 +--------------------+-----------------+--------+-----+
 | Person_ID | Title      | JournalPapers | Age |
 |-----------+------------+---------------+-----|
 | Anna      | PhD_Student| 10            | 24  |
 | Suzan     | PhD_Student| 4             | 25  |
 | Mary      | Student    | 2             | 25  |
 | John      | PhD_Student| 12            | 31  |
```

“Executing” RDF statement

```
:AI_Department :hasAvgYoungDoctStudentsPerformance
```
“Executable Knowledge”
example 3

```
{sapl:l sapl:do java:ubiware.BI_Soft_Library.UnitReportGenerator}
sapl:configuredAs
  { BI:input sapl:is :AI_Department.
    BI:TimeWindowYears sapl:is '3'.
    BI:TimeFinish sapl:is 'now'.
    BI:output sapl:is :BI_ReportURI
  }

:AI_Department :hasBusinessIntelligenceReport exe:Q3
```

```
<http://www.ubiware.com/Al_Department/reports#rep2008-2010>
```
“Executable” Mixed Reality:
Business Intelligence on Top of Linked Data
(concept of Industrial Ontologies Group)
University of Jyväskylä

Contexts for BI services

On-the-fly generated statistics

Executable Focus
Presentation Case for UBIWARE 3.0

User

Linked Data

Capabilities

Linked Data

Capabilities

Special Occasions

- Delight
- Congratulate
- Comfort

User

Linked Data

Capabilities

Linked Data

Capabilities

User
Key Components of UBIWARE
Scientific Impact

1. UBIWARE: Approach and Architecture
2. Engine
3. Language
4. Ontonuts

Business Process Choreography
External Capabilities Orchestration
While the academic and business communities are exited with the new Cloud Computing and SOA slogan: “EaaS: Everything-as-a-Service!”, our group since 2003 is actively working on GUN Computing and Web 5.0, which much more challenging slogan (based on “ψ-projection” technological vision) is: “EaaS4E: EaaS for Everything”, meaning “Really Everything-as-a-Proactive, Semantic and Self-Configurable Web Service Provider and Consumer!”.