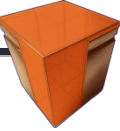


Ambient intelligence: integrating agents, services and ontologies

Dr. Dionisis Kehagias

Senior Research Associate

**Centre for Research and Technology Hellas Informatics and
Telematics Institute**



Purpose

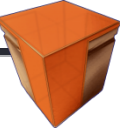
- To provide a gentle introduction to Ambient intelligence (Aml) technology
- To provide a too brief introduction to
 - Software agents
 - Web services
 - Ontologies
- To show how all these technologies fit together in the context of Aml
- To explain how Aml can be implemented through
 - an example of a currently developed Aml project
 - relevant research work conducted @ CERTH/ITI



Presentation outline

- Ambient Intelligence system attributes
- Software Agents
- Web Services
- Ontologies
- Ambient Intelligence integrated systems:
the OASIS case
- Semantic categorisation of Web services
for efficient personalised content provision

Ambient Intelligence



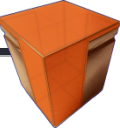
Definitions

- Ambient Intelligence is a **distributed network of intelligent devices** that provides us with information, communication and entertainment.”

Emile Aarts , Rick Harwig, “Ambient Intelligence”

- “Ambient Intelligence is a **network of hidden intelligent interfaces** that recognize our **presence** and **mould** our environment to our immediate needs.”

John Horvath, Telepolis, Making Friends with Big brother

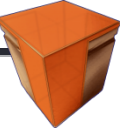


Definitions

- “Ambient Intelligence refers to an exciting new paradigm in information technology, in which people are **empowered** through a digital environment that is **aware** of their **presence** and **context** and is sensitive, adaptive and responsive to their **needs, habits, gestures** and **emotions.**”

Taken from “Ambience Project”

URL: <http://www.extra.research.philips.com/euprojects/ambience>



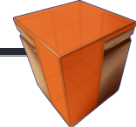
Key-phrases

- Distributed network
- Intelligent Devices
- Recognise our presence
- Context-aware
- Responsive / Adaptive to our needs
- Intelligent hidden interfaces
- Information, communication, entertainment



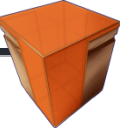
Ambient Intelligence

- **Ubiquitous computing**
 - means the integration of microprocessors into everyday objects like furniture, clothes or toys.
- **Ubiquitous communication**
 - should enable these objects to communicate with each other and with the user
- **Intelligent User Friendly Interfaces**
 - Enables Aml users to **control** and interact with the environment in a natural (e.g. voice) and personalised (user context, preferences) way
- **Security**
 - Seamless (and secure) delivery of services and applications



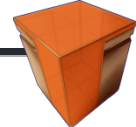
How an Aml environment looks like



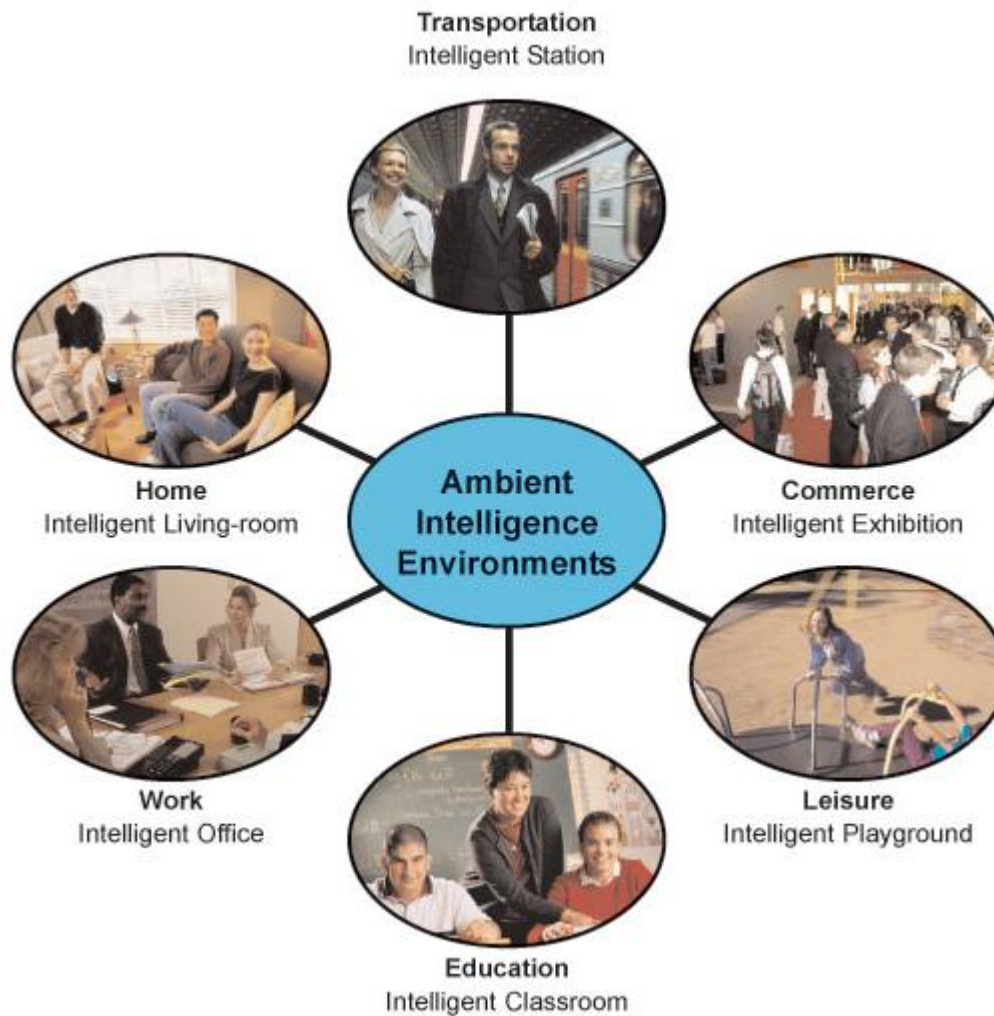


User Context

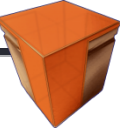
- Any information that can be used to characterise the users and their situation
 - Coming from sensors
 - Temporal and spatial location
 - Environmental attributes
 - Resources nearby
 - Physiological measurements
 - User preferences and profile
 - Schedule, agenda
 - Social context



Examples of Application Domains

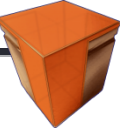


Software Agents



What is an Agent?

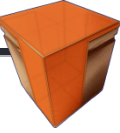
- In general, an entity that interacts with its environment
 - Perception through sensors
 - Actions through effectors or actuators



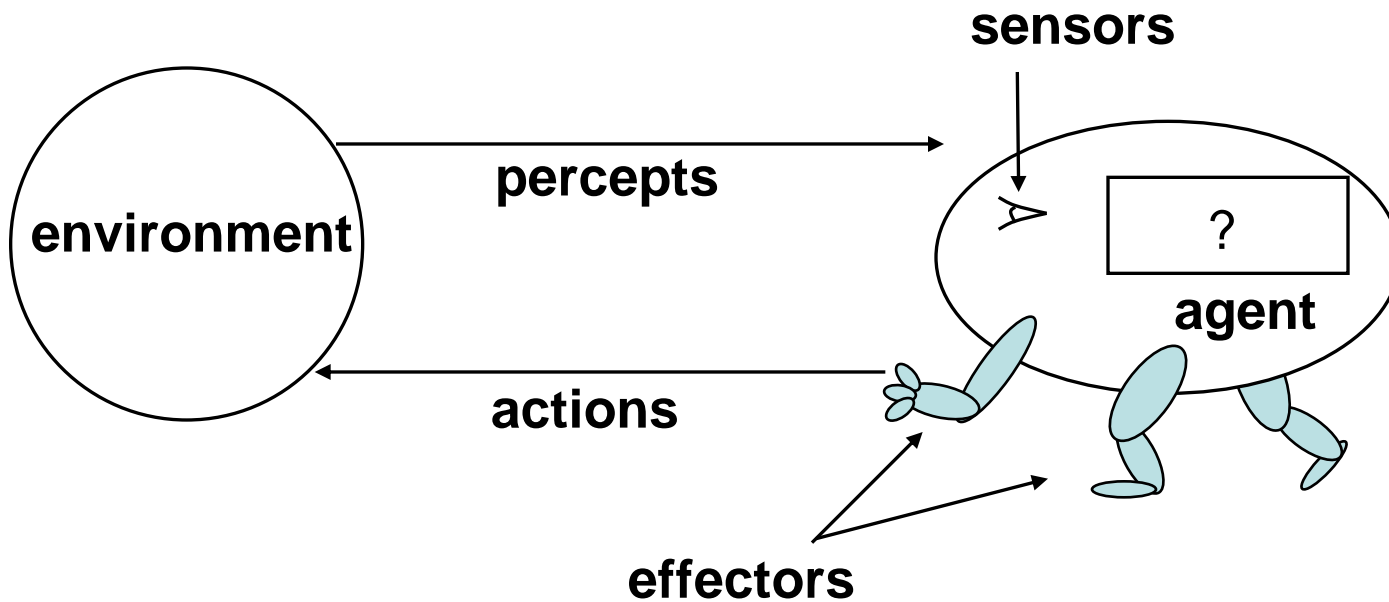
A definition about agents

Autonomous software entities that constantly perform a set of tasks in a goal-oriented way, on behalf of a third-party entity, human or software.

- In real life agents are humans that act on behalf of someone else.
 - ✓ Secret Agents
 - ✓ Travel Agents
 - ✓ Real Estate Agents
 - ✓ Sports/Showbiz Agents
 - ✓ Purchasing Agents
 - *What do these jobs have in common?*
- Software agents are computer programs that act **on behalf of a third-party entity** (human or software program) and employ a set of attributes.



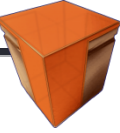
An agent and its environment





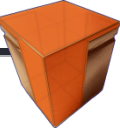
Agent Properties

- Autonomous
 - Take the initiative
- Reactive to the environment
- Proactive
 - Act in a goal-oriented way
- Learning ability
 - Adaptive to the environment
- Social ability
 - Form societies



Examples of software agents

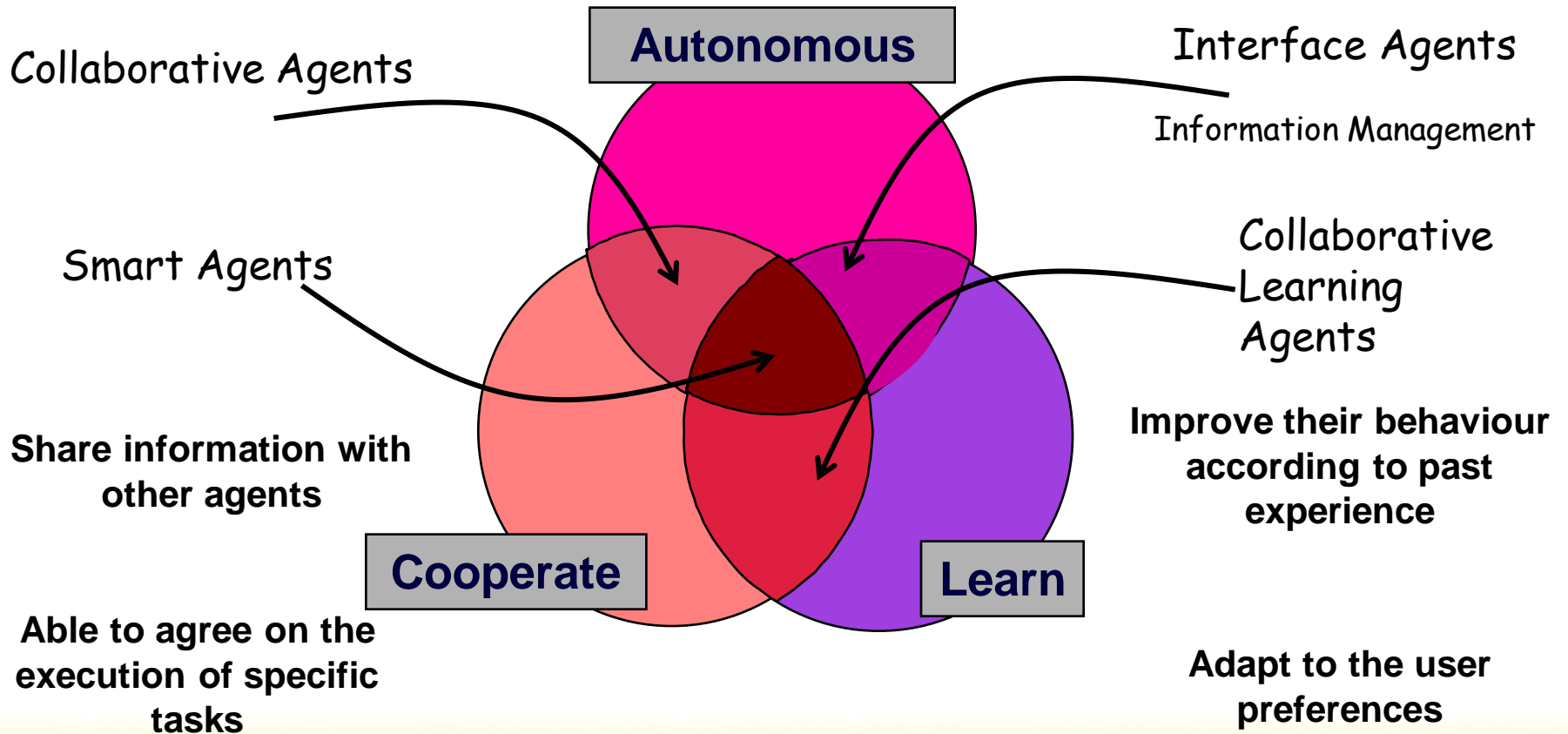
- News-filtering agents
- Shopbots/price comparison agents
- Bidding agents
- Recommender agents
- Personal Assistants
- Middle agents/brokers
- Etc.



An Agent Typology

Do not require supervision

Taking the initiative



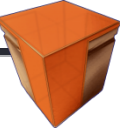
SOURCE: NWANA 1996

Web services



What are Web Services?

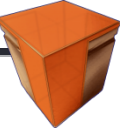
- Applications that can be called remotely via HTTP requests
- Properties of a web service (WS):
 - Runs on a web server
 - Exposes web methods to interested callers
 - Listens for HTTP requests representing commands to invoke web methods
 - Executes web methods and returns the results



WS characteristics

- Language agnostic
- Can be called from any platform or client type
- Use SOAP and XML as the transfer medium
- Allow passing of data through firewalls

- Designed to be consumed by machines



Underlying Technologies

Web Services Stack (Standards)

Directory: Publish & Find Services:	UDDI
-------------------------------------	------

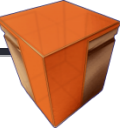
Description: Formal Service Descriptions:	WSDL
---	------

Wire Format: Service Interactions:	SOAP
------------------------------------	------

Universal Data Format:	XML
------------------------	-----

Ubiquitous Communications:	Internet
----------------------------	----------

Simple, Open, Broad Industry Support

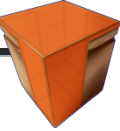


Sample web service

Calc.asmx

```
<%@ WebService Language="C#"
    CodeBehind="~/App_Code/WebService.cs" Class="WebService" %>
using System;
using System.Web.Services;

[WebService (Name="Calculator Web Service",
Description = "Perform simple math over the Web")]
class CalcService
{
    [WebMethod (Description = "Computes the sum of two integers")]
    public int Add (int a, int b) { return a+b;}
    [WebMethod (Description = "Computes the difference between two
        integers")]
    public int Subtract (int a, int b) { return a-b;}
}
```



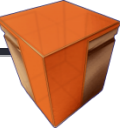
Web Services Description Language - WSDL

If other developers are to consume (that is, write clients for) a Web service that you author, they need to know:

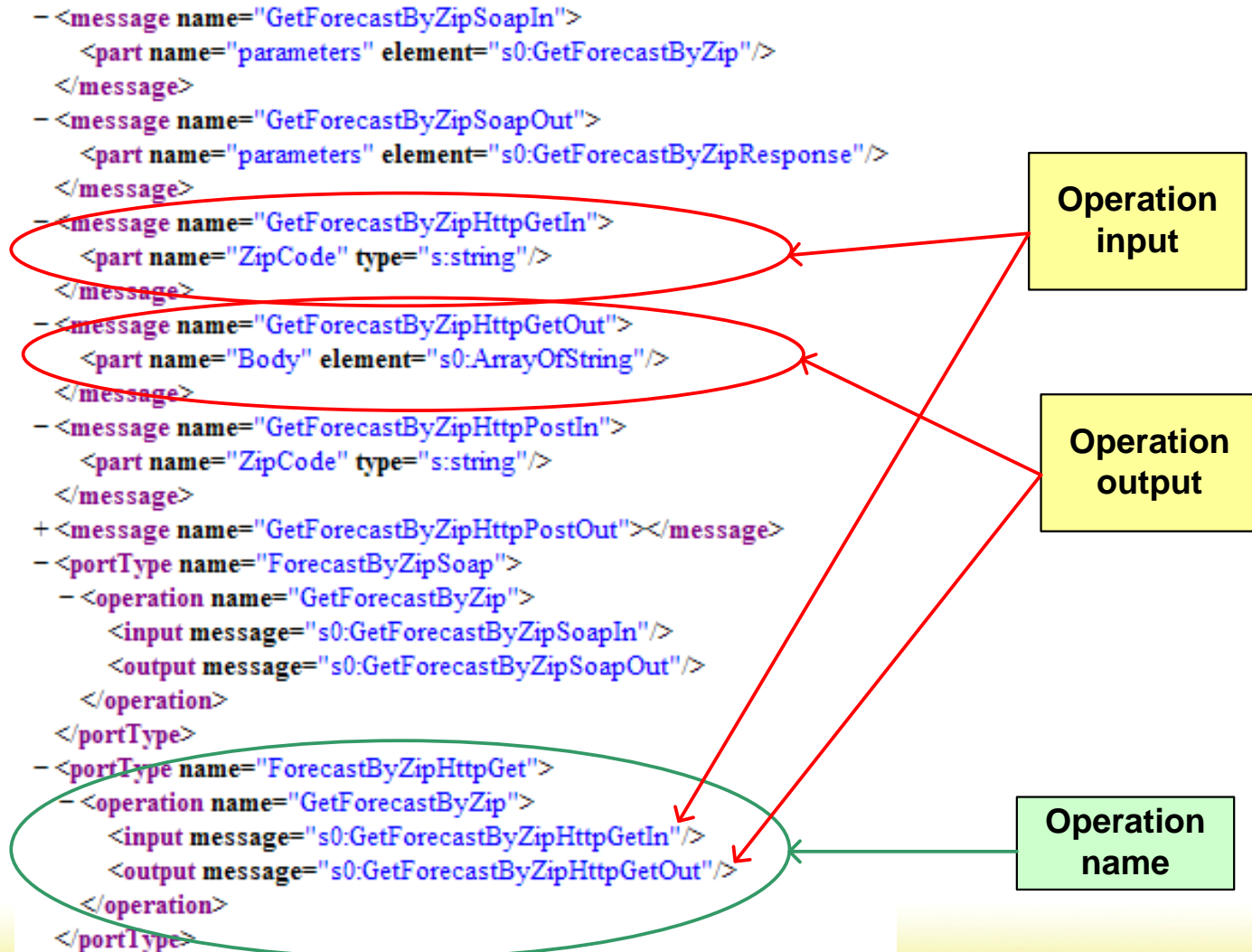
- What web methods your service publishes
- What protocols it supports
- The signatures of its methods
- The web service's location (URL)

All this information and more can be expressed in a language called the **Web Services Description Language** (or WSDL, pronounced 'wiz-dəl').

WSDL is an XML vocabulary <http://www.w3.org/TR/wsdl>.



A sample WSDL file

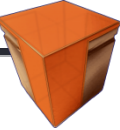


Ontologies



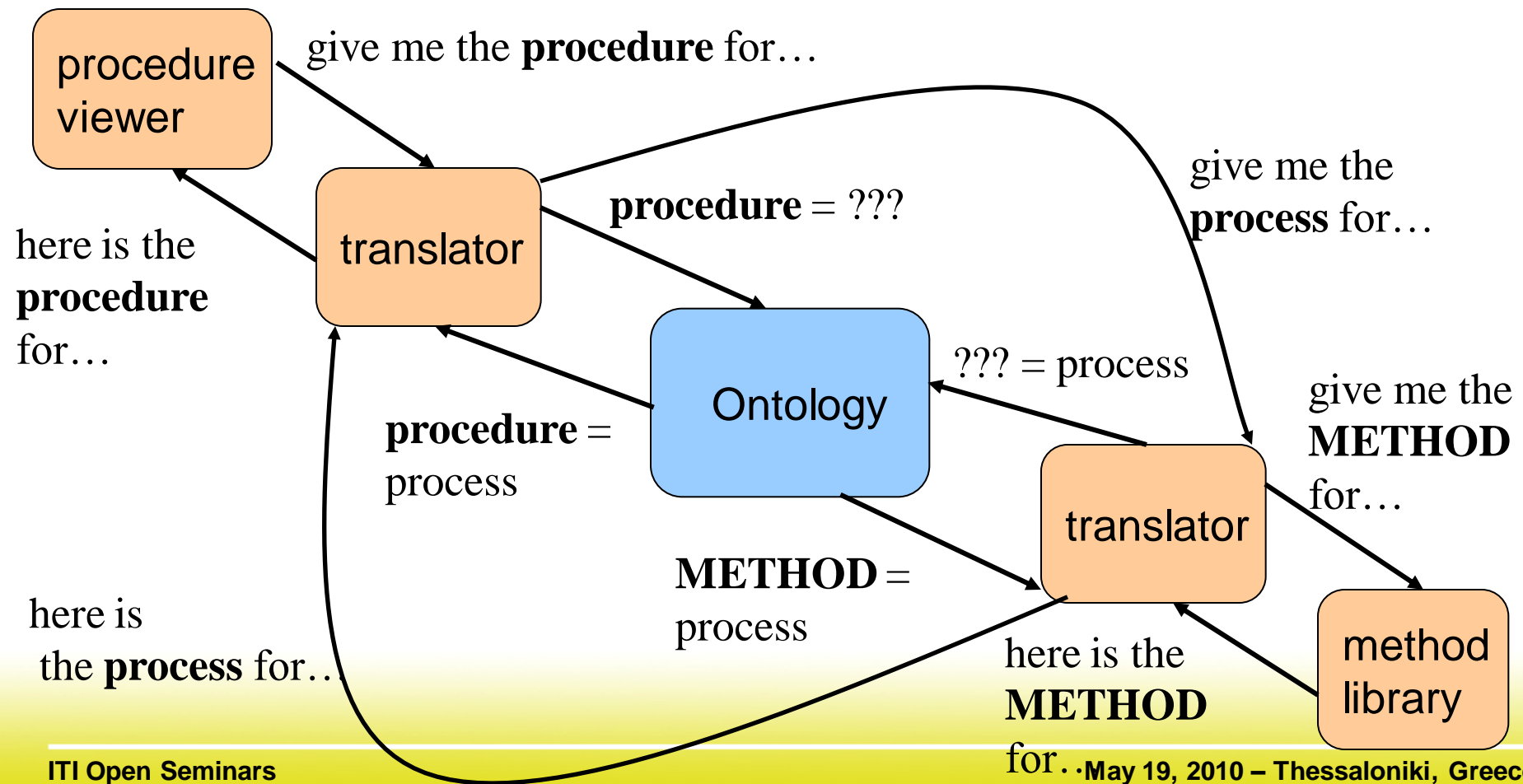
Reasons for using ontologies

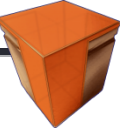
- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit: easier to validate, to change
- To analyze domain knowledge



Example

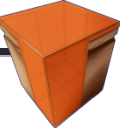
The term ‘procedure’ used by one tool is translated into the term ‘method’ used by the other via the ontology, whose term for the same underlying concept is ‘process’.





Ontologies in practice

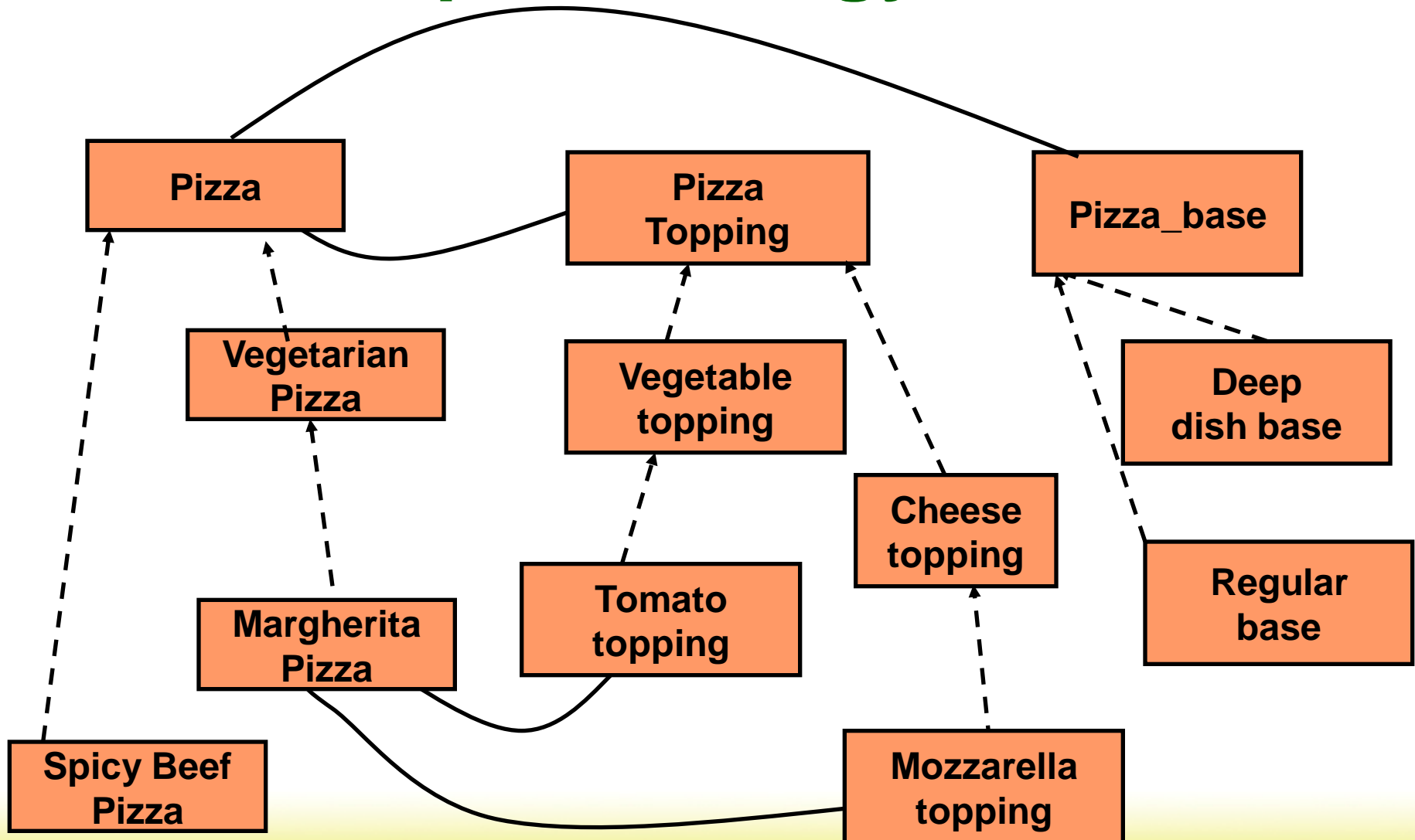
- **Ontology** is a formal explicit description of
 - Concepts in a domain: **classes**, or **concepts**
 - **Subclasses** represent concepts more specific than their superclasses
 - Properties of each concept describing features and attributes of the concept: **slots**, **roles** or **properties**
 - Restrictions on slots: **facets** or **role restrictions**
- A **knowledge base**: an ontology and a set of individual **instances** of classes

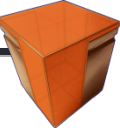


Web Ontology Language OWL

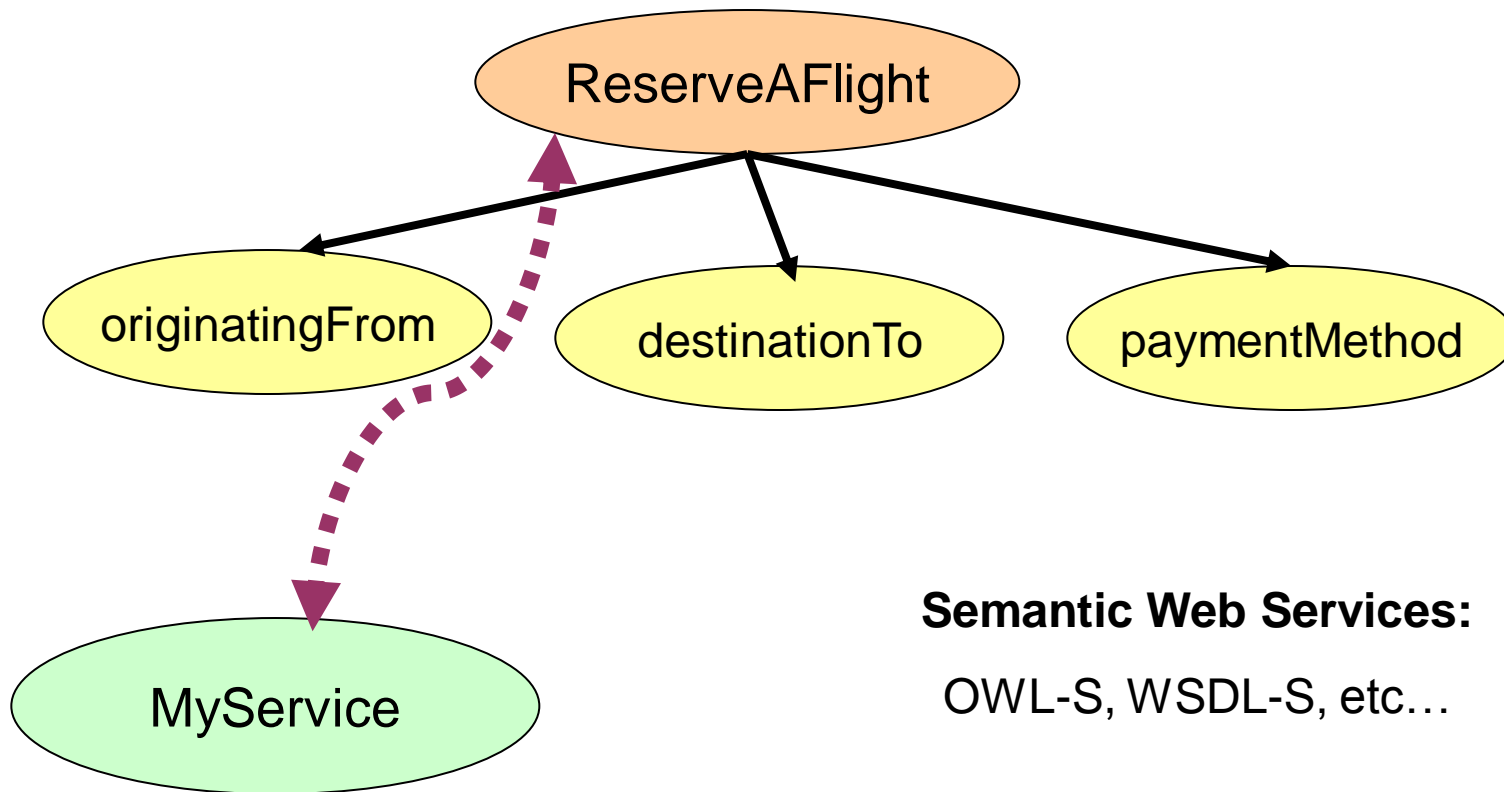
- OWL is a semantic markup language being developed by the World Wide Web Consortium
 - for publishing and sharing ontologies
 - derived from DAML+OIL

A simple ontology: Pizzas





Relating Services with Ontologies



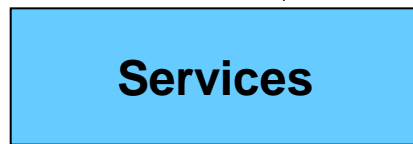
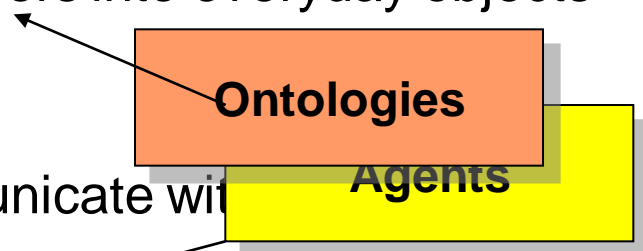
Semantic Web Services:

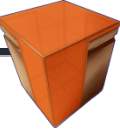
OWL-S, WSDL-S, etc...

Ambient Intelligence Integrated Frameworks

Ambient Intelligence

- **Ubiquitous computing**
 - means the integration of microprocessors into everyday objects like furniture, clothes or toys.
- **Ubiquitous communication**
 - should enable these objects to communicate with the user
- **Intelligent User Friendly Interfaces**
 - Enables Aml users to **control** and interact with the environment in a natural (e.g. voice) and personalised (user context, preferences) way
- **Security**
 - Seamless (and secure) delivery of services and applications





The “OASIS” project

OASIS - a Large Scale Integrated Project partially funded by the European Commission - **aims to develop an open and innovative reference architecture**, that will allow **plug and play and cost-effective interconnection of existing and new services** in all domains required for

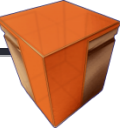
- the independent and autonomous living of older people and
- their enhanced Quality of Life.

Open architecture for **A**ccessible **S**ervices

Integration and **S**tandardisation

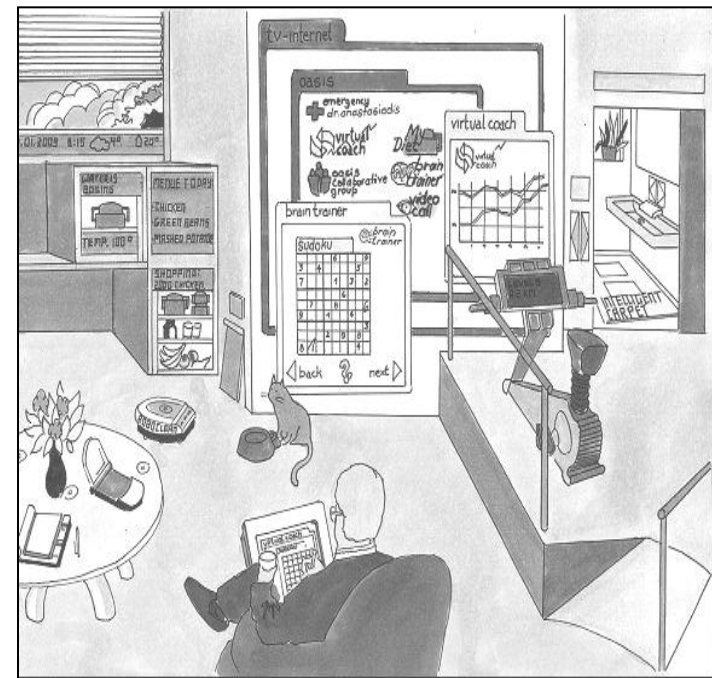
Project data

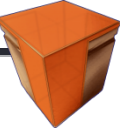
- EC co-funded project (ICT-FP7)
- Coordinator: Philips FIMI (Italy)
- Start/end date: 1 January 2008 - 31 December 2011
- Consortium: 33 Partners from 11 countries
- Project web site: <http://www.oasis-project.eu/>



OASIS Independent Living Applications

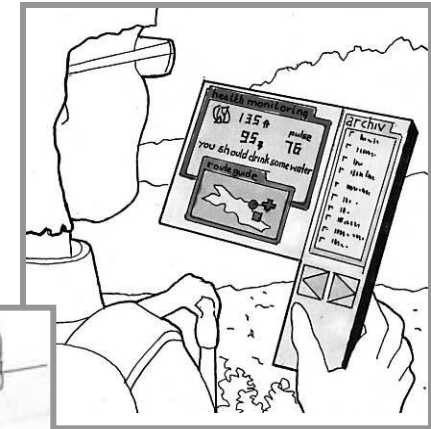
- Nutritional Advisor
- Activity coach
- Brain and skills trainer
- Social communities platform
- Health monitoring
- Environmental Control



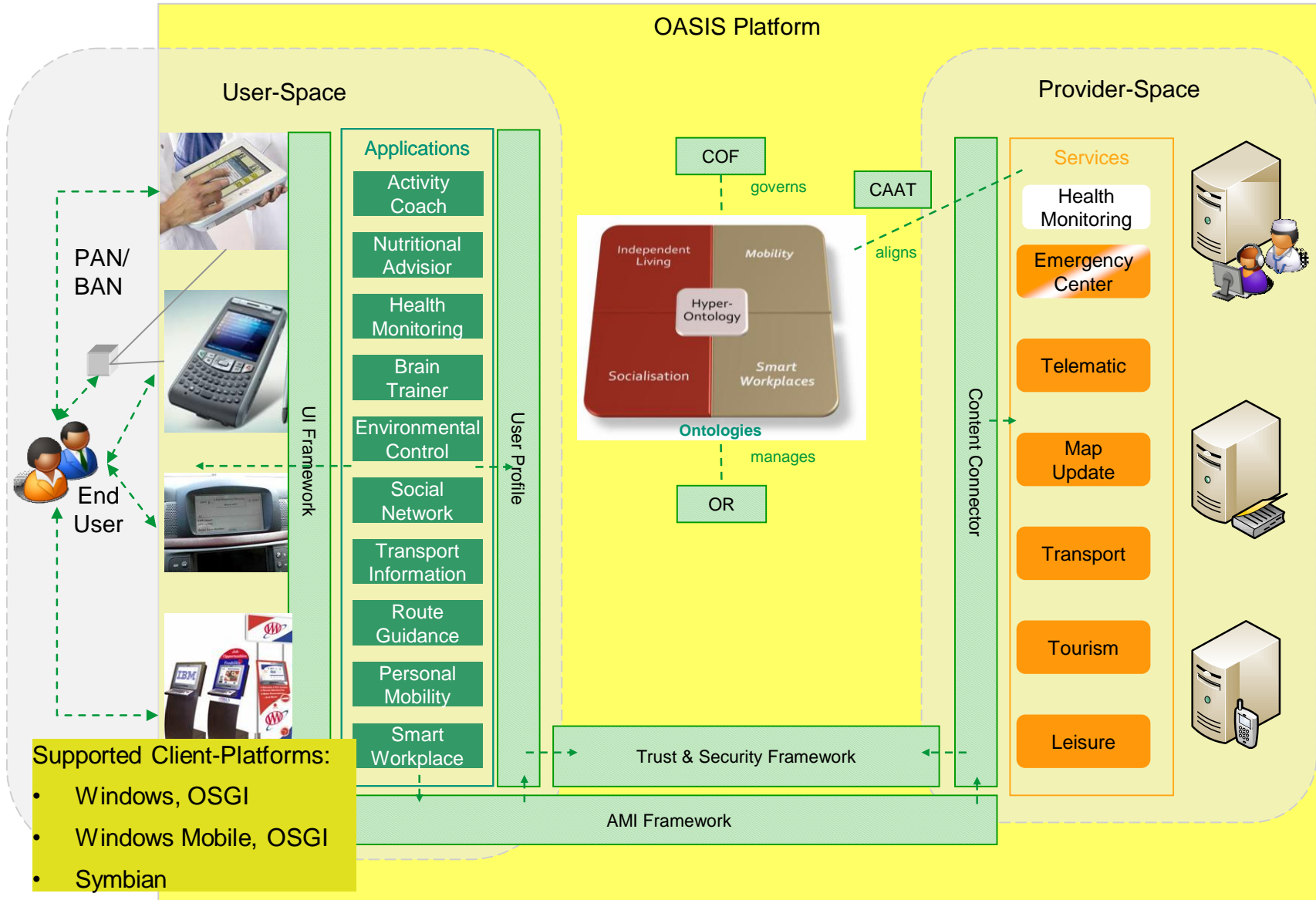


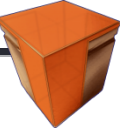
OASIS Autonomous Mobility and Smart Workplaces Applications

- Elderly friendly transport information services
- New, elderly-friendly route guidance
- Personal mobility
- Smart workplaces applications



Conceptual Architecture





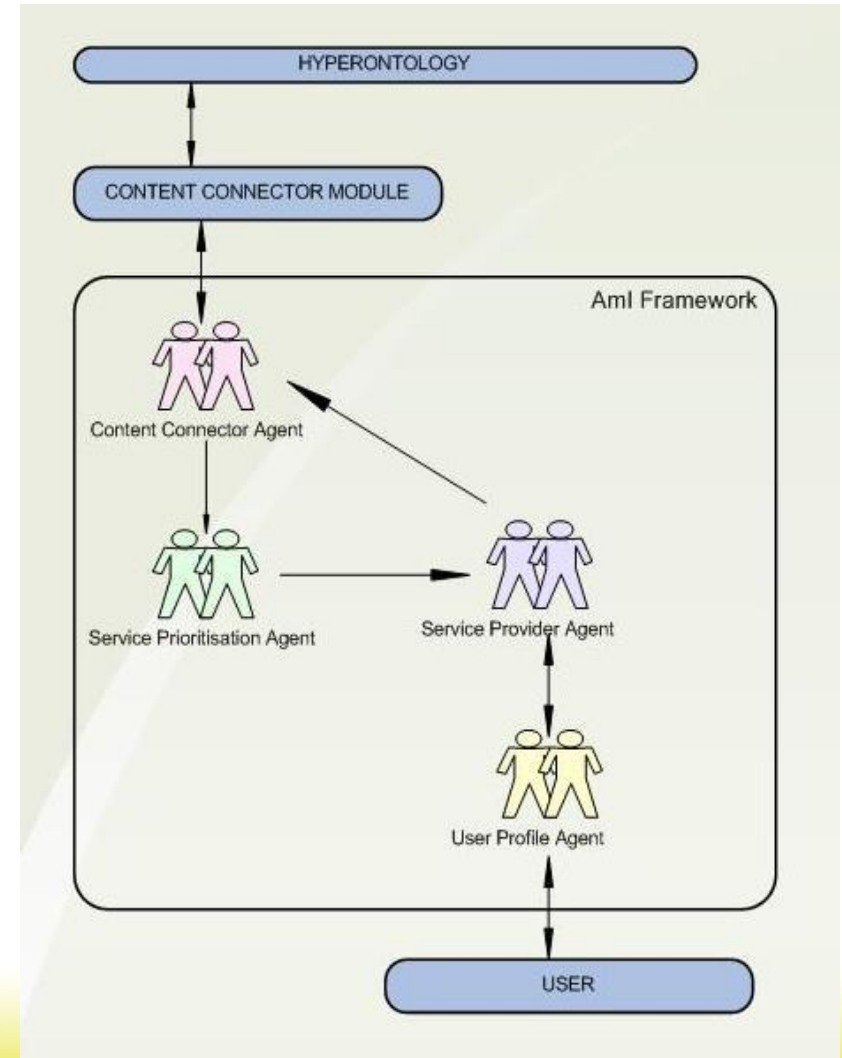
“Hyper-Ontology”

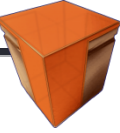
- Has a formal specification
 - Machine process-able
 - Consists of a collection of primitive domain ontologies
- Define shared conceptualisations
 - Captures consensual knowledge
 - Describes services and devices
- Enable knowledge sharing in an open and dynamic distributed environment



OASIS Agent Platform

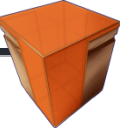
- User Profile Agents
- Service Provider Agents
- Service Prioritisation Agents
- Content Connector Agents



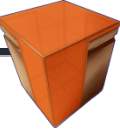


Exploiting User Context for Web Service Discovery

- For selecting services in a context-sensitive manner
 - Services should be discovered based on their semantic descriptions
- Agents should
 - Query the context of the user
 - Maps the preferences of the user with the properties of the services advertised



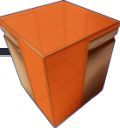
Semantic Categorisation of Web Services for Personalised Content Provision



Semantic categorization of WS

Problem definition

- How to semantically recognize a web service (WS) by its structural elements.
 - Semantic categorization of a WS into application domains
 - Semantic categorization of WS operations into “ideal” operations defined in terms of an ontology
- Web service categorization is important for semantic annotation of services.
- This helps the dynamic creation of service catalogues and facilitates service search and discovery.



3-Layer Semantic Categorization

➤ 1st Layer

- WS is classified into one **domain**

➤ 2nd Layer

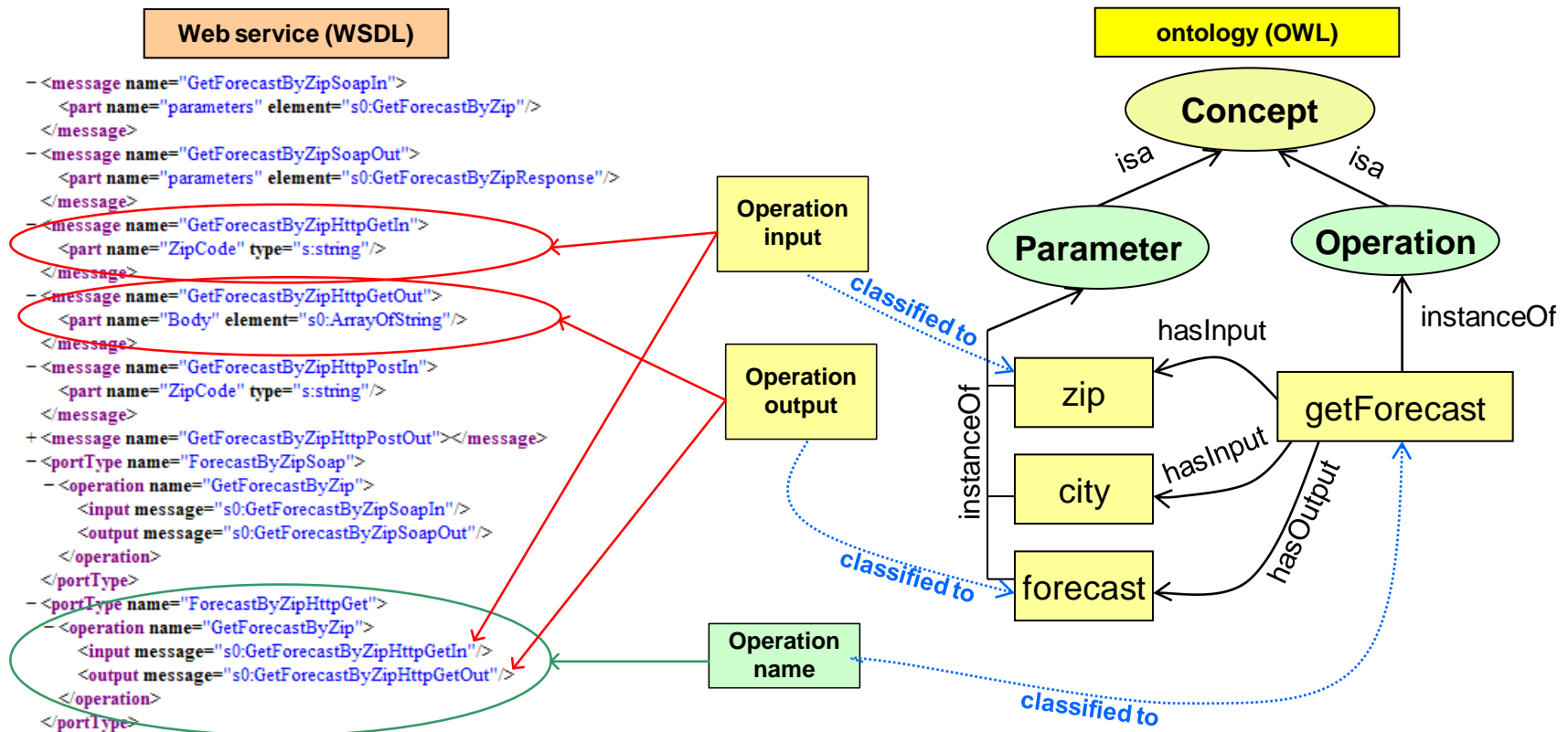
- WS operations are classified into the best matching ontology operations
("ideal operations")

➤ 3rd Layer

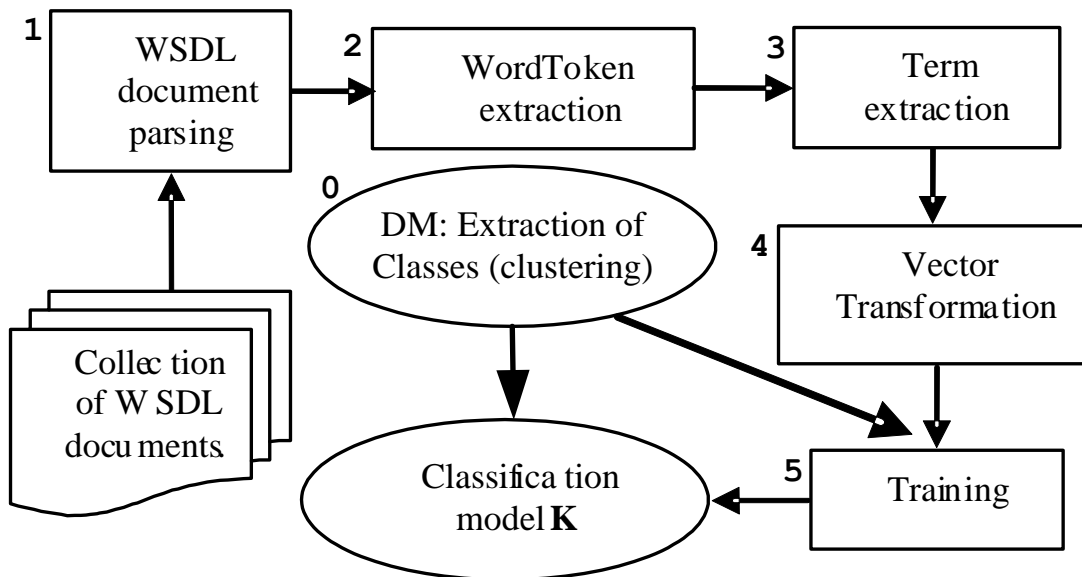
- WS operation **input and output parameters** are classified into their ontologically defined counterparts (parameters of "ideal" operations)

Web service classification

In the 2nd and 3rd layers our goal is to classify WS operations and their i/o parameters with respect to the ontologically defined counterparts



Semantic Categorization Technique (1st and 2nd layers)



1st (2nd) layer classification

problem:

m web services
(operations) to be
classified into n domains
("ideal" operations).

$$A = \{a_1, \dots, a_m\}$$

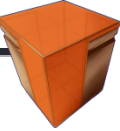
$$D = \{d_1, \dots, d_n\}$$

$$K = A \cdot D$$

$$k_{ij} = \begin{cases} 1, & \text{if } a_i \text{ belongs to } d_j \\ 0, & \text{otherwise} \end{cases}$$

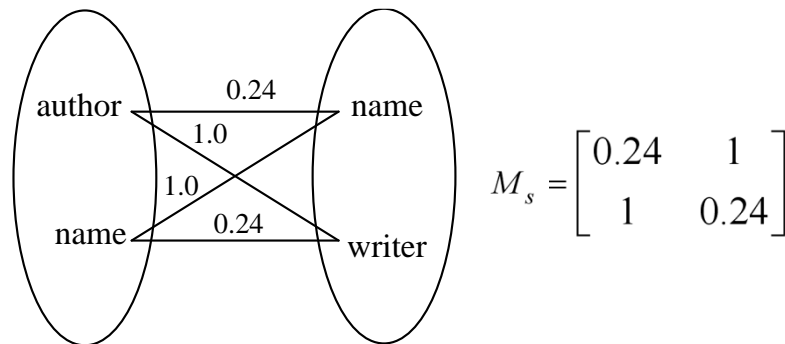
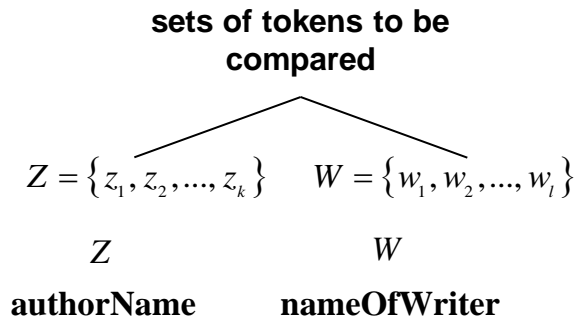
The semantic categorization procedure (1st layer)

1. WSDL elements are extracted via a parser.
2. Extracted elements are tokenized
3. Specific terms are extracted
4. Vectors are defined to represent WS operations $v = [v_0, v_1, \dots, v_k, c]$
5. Training of a classifier is conducted based on vector data.



Categorization of input/output parameters (3rd level)

Since the third layer does not contain adequate information, as opposed to the previous two layers, a different classification method was required.



- Each name of I/O operation parameter is compared to all ideal operations I/O.
- The algorithm uses three levels of matching, lexicographic, structure and data type matching.

3 score matrices: M_s, M_d, M_g

Overall score matrix: $M = w_1 M_s + w_2 M_d + w_3 M_g$

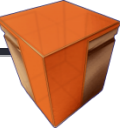
$s \rightarrow$ structure, $d \rightarrow$ data type, $g \rightarrow$ lexicographic

- Lexicographic similarity is computed using *WordNet:Similarity*, and *n-grams* model.



Evaluation of the WS evaluation accuracy

- A set of experiments were conducted in order to:
 - Evaluate the performance of the WS mining classification approach
 - See the impact of each one of the tunable parameters
 - Compare the WS mining mechanism with a known approach

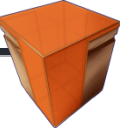


Evaluation data

➤ A set of publicly available WS were used in our evaluation process from the following repositories:

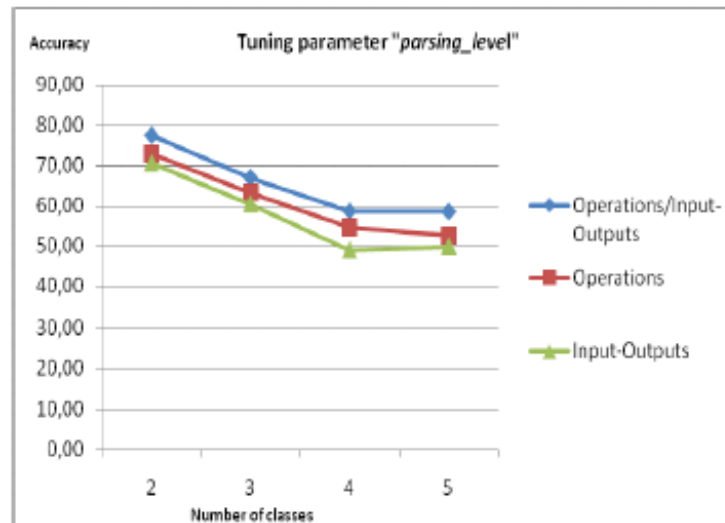
- WebServiceX.NET
- XMethods
- Seekda!
- random WS

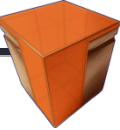
Domain	Number of Web services
Business and Money	98
Tourism and Leisure	21
Communication	68
Geographic	79
Total	266



Evaluation process

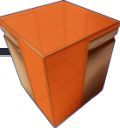
- We varied the value of the parsing level parameter while keeping the rest of the parameters constant





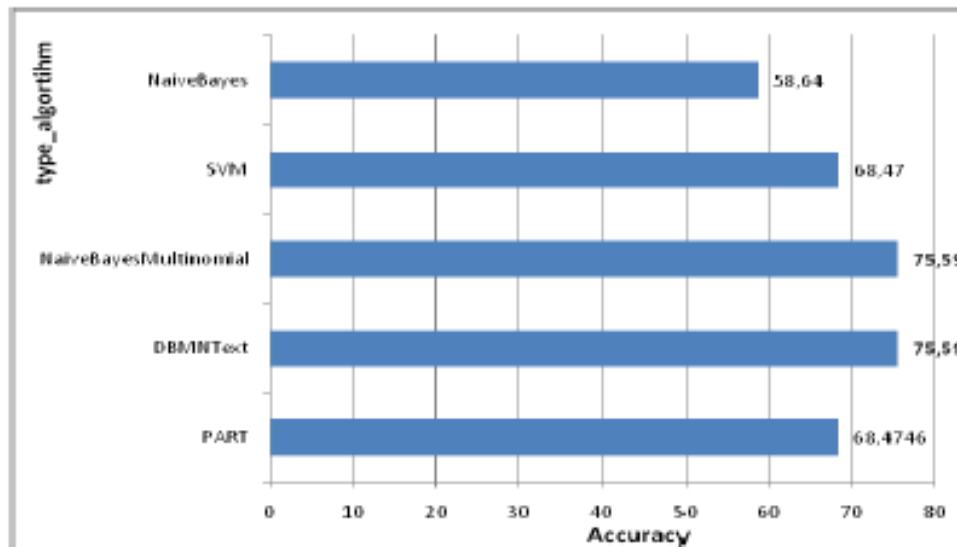
Tuning classification algorithms

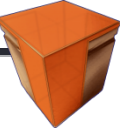
- The goal of this experiment is to identify the impact of each algorithm
- We consequently applied the following algorithms:
 - Naïve Bayes,
 - support vector machines (SVM),
 - Naïve Bayes Multinomial,
 - Discriminative Multinomial Naïve Bayes (BDMNText)
 - PART algorithm, (a rule-based classifier).



Results

- Best performance:
 - Naïve Bayes Multinomial,
 - Discriminative Multinomial Naïve Bayes (BDMNText)





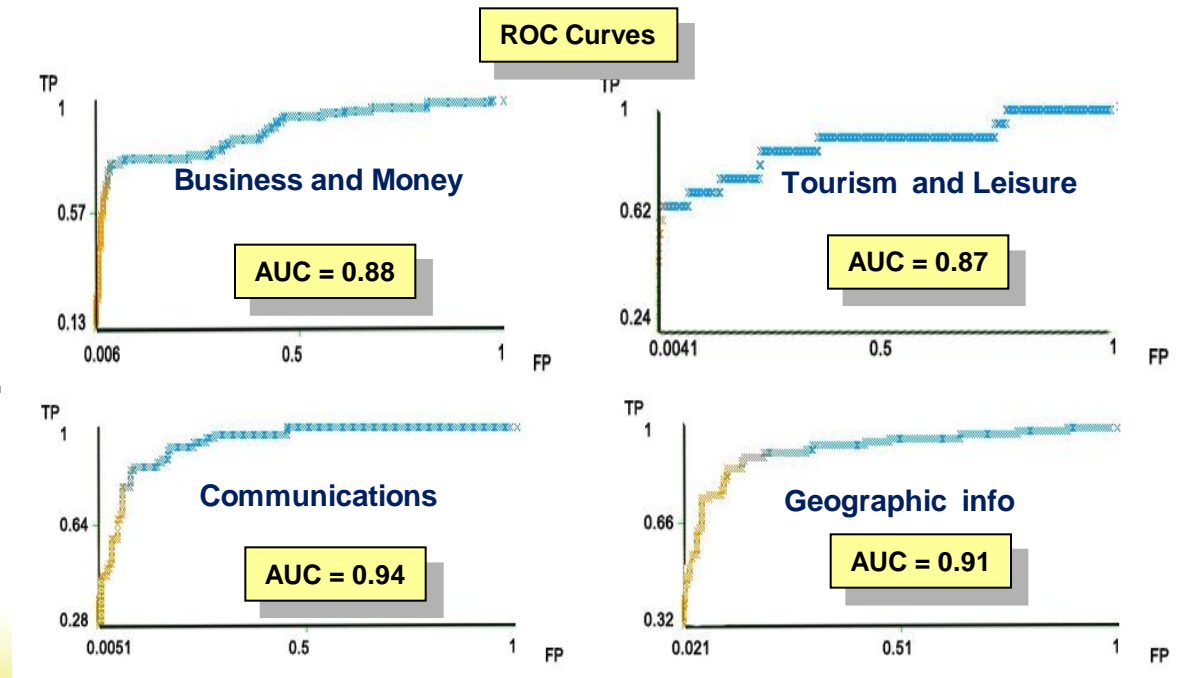
Evaluation: 1st layer

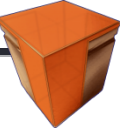
Domain	Number of Web services
Business and Money	98
Tourism and Leisure	21
Communication	68
Geographic	79
Total	266

- The area under the ROC curve (AUC) measures the discriminating ability of a classification model.
- The larger the AUC becomes, the more accurate the model.

➤ A ROC curve plots true positive (TP) vs. false positive (FP) rates.

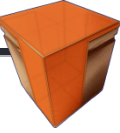
Average classification accuracy: 77.44%



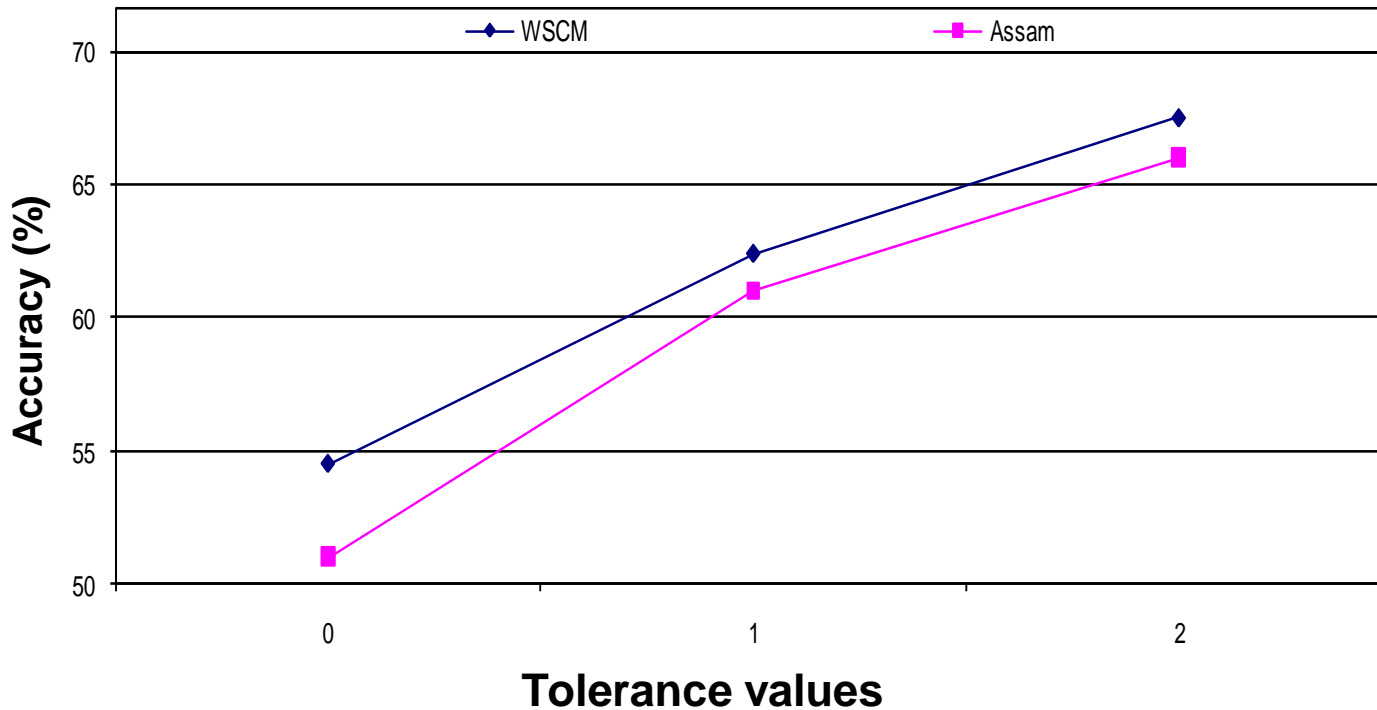


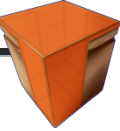
Evaluation: 2nd layer

Domain	Accuracy (%)
Business and Money	62.01
Tourism and Leisure	60.47
Communication	65.05
Geographic	73.95



Comparison with ASSAM (1st layer)



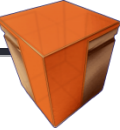


Conclusions and Future Work

- **The proposed categorization mechanism shows efficient accuracy especially for low tolerance values.**
- **This is proven by the previous comparison.**

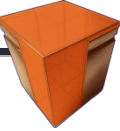
Future plans:

- **Comparison with additional tools.**
- **Further improvement of the classification accuracy.**



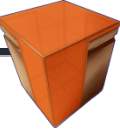
Summary and concluding remarks

- Aml is a new IT paradigm that is aware of user presence and context
- Software Agents are ideal candidates to
 - handle responses from the ubiquitous computing environment (equipped with sensors)
 - and provide intelligence user interaction (equipped with effectors)
- Web services are computer programs that accessible on the Internet through a standards-abiding and secure way
 - Ideal to provide secure content provision in an Aml



Summary and concluding remarks

- Ontologies are a knowledge representation paradigm capable for ubiquitous communication and share of knowledge
 - Provide a common understanding of the supported application domains and the services/devices to be provided
 - They also enable semantic web services
- The OASIS project provides an implementation of an Aml that combines all three technologies



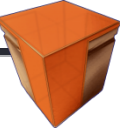
Summary and concluding remarks

- Seamless service integration is facilitated by semantic categorisation of web services
 - Ontologies play a significant role here
- Substantial research is conducted on the improvement of the categorisation accuracy
- Web service categorization is important for automatic semantic annotation of services
- This helps the dynamic creation of service catalogues and facilitates service search and discovery



Summary and concluding remarks

- The evolution of Aml relies on how a set of involved technologies will evolve
 - **Industrial support:**
 - Web services
 - Wireless networks
 - Ubiquitous computing – wearable devices
 - End user mobile devices – smart phones
 - **Research-oriented**
 - Software agents
 - Ontologies



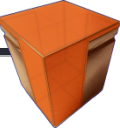
Contact info

- Dr. Dionisis Kehagias

diok@iti.gr

- Informatics and Telematics Institute

<http://www.iti.gr>



Q&A