Ref. Ares(2015)334123 - 28/01/2015

PROJECT FINAL REPORT

Grant Agreement number: 288385

Project acronym: Internet of Things Environment for Service Creation and Testing

Project title: IoT.est

Funding Scheme: STREP

Period covered: from 1st October 2011 to 30th September 2014

Name, title and organisation of the scientific representative of the project's coordinator:

Professor Klaus Moessner, University of Surrey

Tel: +44 1483 683468

Fax: +44 1483 686011

E-mail: K.Moessner@surrey.ac.uk

Project website address: http://ict-iotest.eu/iotest/

Document Ref.	D1.7
Workpackage	1
Lead Contractor	US
Editor	Chris Alais, Suparna De, Lara Lopez
Contributing Authors	All partners
Planned Delivery Date	30.11.14
Actual Delivery Date	27.01.15
Dissemination Level	Public
Status	Final – after EC recommendations
Reviewed by	Consortium

Contents

List of Figures

Final Publishable Summary Report

Executive Summary

Despite much work to break up the silos and research into more generic, cross-domain mechanisms, most implementations of Internet of Things (IoT) architectures are still confined to particular application areas and tailored to meet only the limited requirements of their specific applications. Domain restricted solutions are still used in the individual sectors and this still restricts both uptake and penetration of generic Internet of Things features and functions (and services) that can be tailored into domain specific applications; this applies in particular to innovative business processes. The provisioning of IoT enabled business services (in short IoT services) is, in general, a time- and cost extensive process. Data acquisition, quality control, context interpretation, decision support, and action control are complex processes a fact that is even more problematic when considering that they rely on often unreliable data sources (and actuation). This poses a challenge that applies to both object-to-object (O2O) and object-to-person (O2P) oriented services. As a result, opportunities remain often unused in today's Internet-connected objects platforms.

To overcome technology & sector boundaries, be able to dynamically design and integrate new types of services and generate new business opportunities requires a **service creation environment** that goes beyond the mere definition, design, instantiation and deployment of services. A service creation environment for IoT services must be able to gather and exploit data and information from sensors and actuators that use different communication technologies/formats. It must be able to facilitate service provision of services that are robust against changes and communication issues that may arise when sensors and actuators are not always available. Looking into the domain and service developer community, there are several sophisticated Service Creation Environments (SCE) which support application developers in rapid service creation and deployment in different platforms. Main shortcoming of these service creation environments is that they have mostly not been developed for the IoT where a large number of resources and the need of automated interpretation of environmental and context information needs to be considered. To accelerate the introduction of new services, effective service creation environment architecture needs to be dynamic and needs to provide a set of features that go beyond the usual SCEs' capabilities, they include:

1. Orchestration, i.e. composition, of business services based on re-usable IoT service components,
2. Self-management capable components for automated configuration and testing of services for "things"
3. Abstraction of the heterogeneity of underlying technologies to ensure interoperability

Beyond this, the mobility of objects and the multiplicity of contexts in which an application may be delivered make the development and maintenance of services an error prone challenge. To overcome this, IoT.est has integrated self-testing capabilities in the service development and maintenance from the very early stages of the service life cycle. The project did investigate and developed a systematic approach that determines how formal test procedures need to be integrated in the service development, and in particular in the service creation environment.

IoT.est did establish mechanisms and tools that ease the creation and provision of IoT enabled business services by bringing together the three disciplines Internet of Things, Service Engineering and Testing.

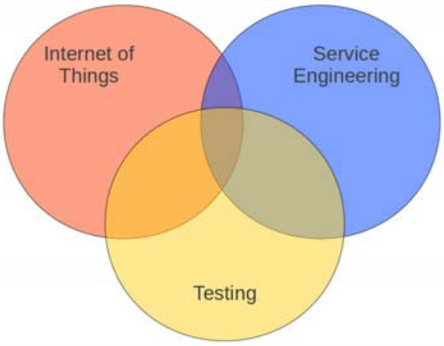


Figure 1 – Integrated approach of IoT.est

1 Project Context and Objectives

The main objective of IoT.est was:

> ***"To develop a service creation and provision environment for reliable "Internet of Things" enabled business processes"***

The project developed and demonstrated an IoT service creation environment, which helps bridge the gap between business services and the heterogeneity of networked sensors, actuators and objects. To cope with the dynamic IoT environments and to ensure reliable services the project did implement a systematic approach to integrate testing into all service life cycle phases.

The project distinguished four service life cycle phases belonging either to design-time or run-time. IoT.est defined design-time as *service creation time* that splits into modeling and development phases. While the run-time aspects cover service provision that can be subdivided into service deployment and service execution. Each phase of the service life cycle is supported by the corresponding test and monitoring phase as listed below and shown in Figure 2:

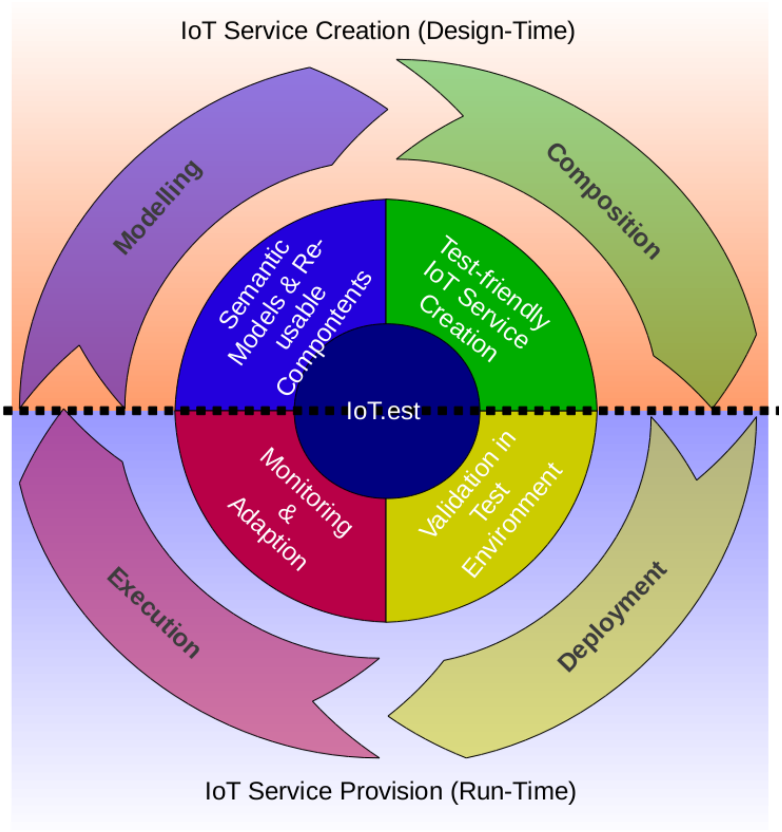


Figure 2 – Life-Cycle Management for IoT enabled Business Processes

Service Creation (Design-Time Phase):

- Service Modeling
  - Syntax and semantic for IoT enabled business processes and it's re-usable components
  - Semi-automatically and semantically driven IoT process composition:
    - Re-use of libraries of tested IoT service components (e.g. data acquisition and distribution, pattern recognition, context interpretation, human behavior recognition, decision support, and (business) action control)

    - Data flow mapping generation.
    - Service process adaptation at design time, based on domain specific contextual information.
- Tools for testing of model consistency
    - Checking data flow, semantic compatibility and consistency.
- Service Development
    - SCE with re-usable components for data acquisition and distribution, pattern recognition, context interpretation, human behavior recognition, decision support, and (business) action control.
- Tools for component and system testing.
    - Automated generation of Test suites for testing service characteristics (i.e. reliability, adaptability, etc.).

Service Provisioning (Run-Time Phase):

- Service Deployment
    - Tools for automated deployment of service in a distributed heterogeneous environment.
    - Automated packaging and deployment on private virtualized environments.
- Service Execution
    - Adaptive capabilities of re-usable components for data acquisition and distribution, pattern recognition, context interpretation, human behavior recognition, decision support, and (business) action control.
- Performance monitoring and service adaptation and personalization.
    - Tools for monitoring the environment context.
    - Run-time IoT process adaptation based on monitored context, service requirements, employing parameterized service components and flow adaptation based on event processing.

IoT.est provides a framework for efficient and reliable service creation and provision by integrating self-testing and self-adaptation in all service life cycle phases. It addressed four key issues related to the different service life cycle phases:

1. Research methods to *derive semi-automatically services and related tests* from semantic service descriptions based on standard service interfaces and re-usable service and test components.
2. Integrate testing into a Service Creation Environment supporting incremental service evolution by *regression tests*. When adding new functionalities, the service components and system tests have been included to ensure backward compatibility with previous service releases.
3. Definition of a framework for service *validation tests* before deployment in the service provider’s infrastructure, including automated deployment procedures based on semantics for service resource requirements and network capabilities.
4. Development of run-time *monitoring* mechanisms which enable service adaptation to environment changes and to adjust network parameters (e.g. Quality of Service).

The project did prototype and demonstrate its major concepts and evaluated the results for exploitation towards future service creation, deployment and testing products. The test-driven SCE has be validated in business scenarios, such as home/house automation.

To achieve these challenging objectives the project has brought together a well-focused consortium of industry and academia with expertise in Internet of Things, service creation and provision and testing. The partners did build upon a long record of developing service creation environments in former projects and designing service testing products.

In order to achieve the project objectives, the project was structured into 7 Work Packages (WP) as illustrated in 3.

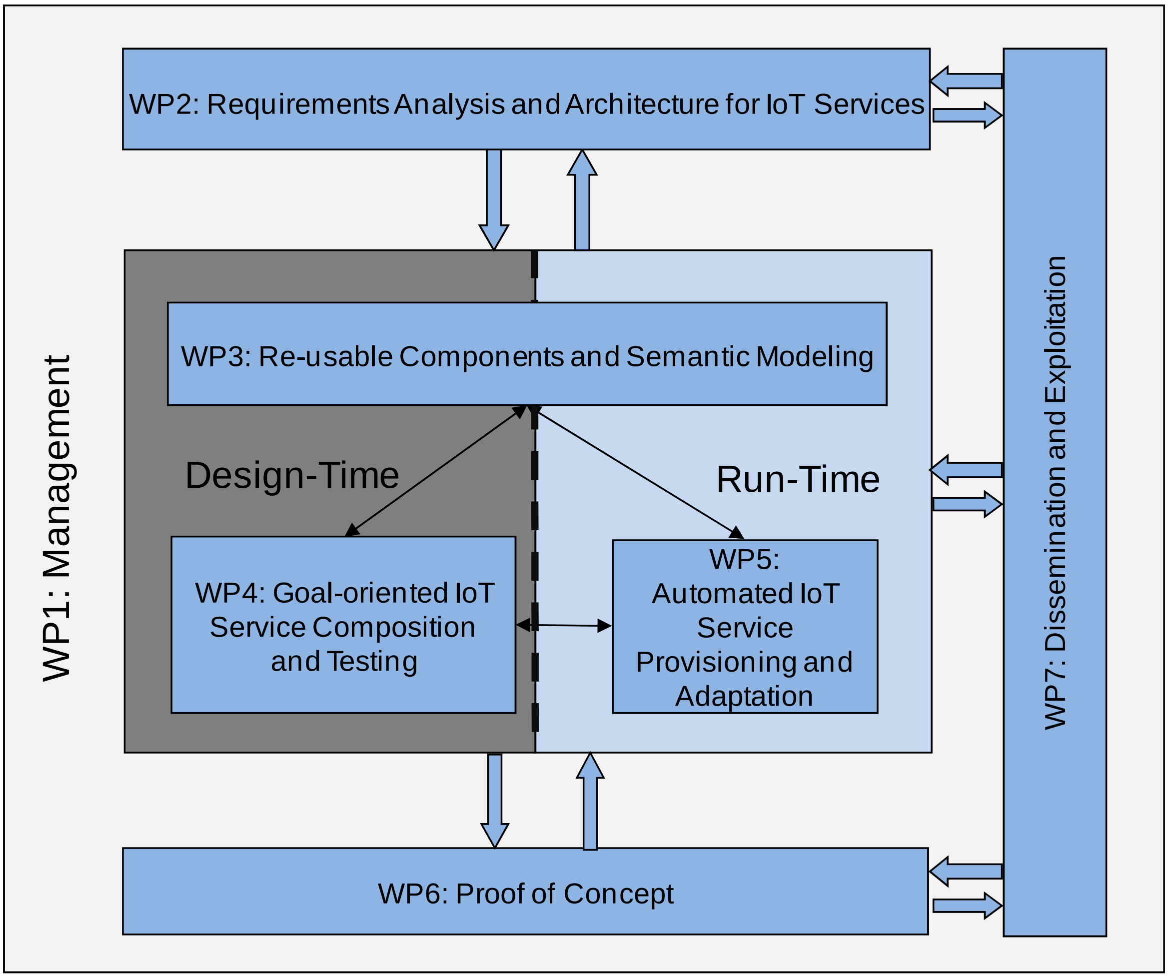


Figure 3 – Overall Structure of the Work Packages

All inclusive activities of project organisation and management are performed in WP1. Three work packages (WP3, WP4 and WP5) form the technical core of the project and are driven by requirement and scenario analysis, and framework definition in WP2. Overall project results are acquired and evaluated in WP6. The dissemination of research concepts and exploitation of results derived from all work packages were conducted in WP7. Each work package included several activities as listed in the following:

- **WP1: Project Management**
  This work package included activities performed by all management bodies in the project. It covers administrative, financial and project technical and operational management tasks. The activities in this work package are listed in the following.

  - Activity 1.1: Management & Quality Control
  - Activity 1.2: Legal, Administrative & Financial Management
  - Activity 1.3: Coordination & Concertation

- **WP2: Requirements Analysis and Architecture for IoT Services**
  This work package focuses on defining and developing the overall concepts of the project. It focused on design rules and a framework for test friendly IoT service life cycle management. The activities in this work package are listed in the following.

  - Activity 2.1: Scenarios and requirements
  - Activity 2.2: Global reference architecture
  - Activity 2.3: IoT service life cycle framework

- **WP3: Re-usable Components and Semantic Modelling**
  This work package concentrated on developing the semantic description framework. Moreover it derived and developed re-usable adaptive service components from analysis of various IoT enabled business processes. Based on IoT service requirements the work package did derive and develop typical re-usable test components. The activities in this work package are listed in the following.

  - Activity 3.1: Semantic description framework for IoT services
  - Activity 3.2: Derivation of re-usable service components from business scenarios
  - Activity 3.3: Query Interfaces, Discovery and Directory Services
  - Activity 3.4: Development of re-usable test components and test interfaces for IoT services

- **WP4: Goal-oriented IoT Service Composition and Testing (Design-Time)**
  This work package focused on the development of a test-friendly service creation environment. It did develop a knowledge based approach to compose re-usable service components and integrated systematically testing into the service creation. The activities in this work package are listed in the following.

  - Activity 4.1: Knowledge-based composition of IoT services
  - Activity 4.2: Inference of SUT model from semantic IoT Service Description
  - Activity 4.3: Knowledge-based inference for test suite from SUT model
  - Activity 4.4: Test integration into IoT service composition

- **WP5: Automated IoT Service Provisioning (Run-Time)**
  This work package provided semantic descriptions for service resources and network capabilities. Automated service provision and support for detecting the changes and making decisions for service adaptability were also covered in this work package. A test environment did ensure testing of the IoT service before deployment in the life system. The activities in this work package are listed in the following.

  - Activity 5.1: Semantics for service resource requirements and network capabilities
  - Activity 5.2: Testing environment
  - Activity 5.3: Automated large scale deployment
  - Activity 5.4: Performance monitoring and service adaptation

- **WP6: Proof of Concept**
  This work package provided prototypes and applications to test and verify the architecture, designed components and methods. The activities in this work package are listed in the following.

  - Activity 6.1: Scenario Specification
  - Activity 6.2: Development, Integration and Validation
  - Activity 6.3: Overall Evaluation

- **WP7: Dissemination and Exploitation**
  This work package focused on exploitation and dissemination activities, including scientific and research publications, participation in workshops, contributions to standardisation and final exploitation plans for the project results. The activities in this work package are listed in the following.

  - Activity 7.1: Dissemination of Results
  - Activity 7.2: Report on, and Contribute to Standardisation
  - Activity 7.3: Exploitation of Results

2 Main Scientific and Technical Results and Foreground

2.1 Architecture for IoT Services

The architecture design extends the existing architectures on Internet of Things and takes the unique characteristics of IoT based services into consideration. The knowledge management component in the architecture is developed based on a semantic description framework, which provides a foundation for knowledge based service discovery, composition and testing. The architecture contains an IoT service composition environment with re-usable components for a number of common functionalities such as data integration and management, sensor data mining, and pattern extraction. Employing those re-usable components facilitates service creation and testing. Another notable feature of the architecture is that it provides mechanisms to test IoT services in the service providers' infrastructure before deployment. Furthermore, the service runtime environment contains service monitoring mechanisms for context-aware service adaptation which is able to respond to changes in the environment. The developed architecture serves as a reference framework for the IoT.est project and other IoT enabled services and applications.

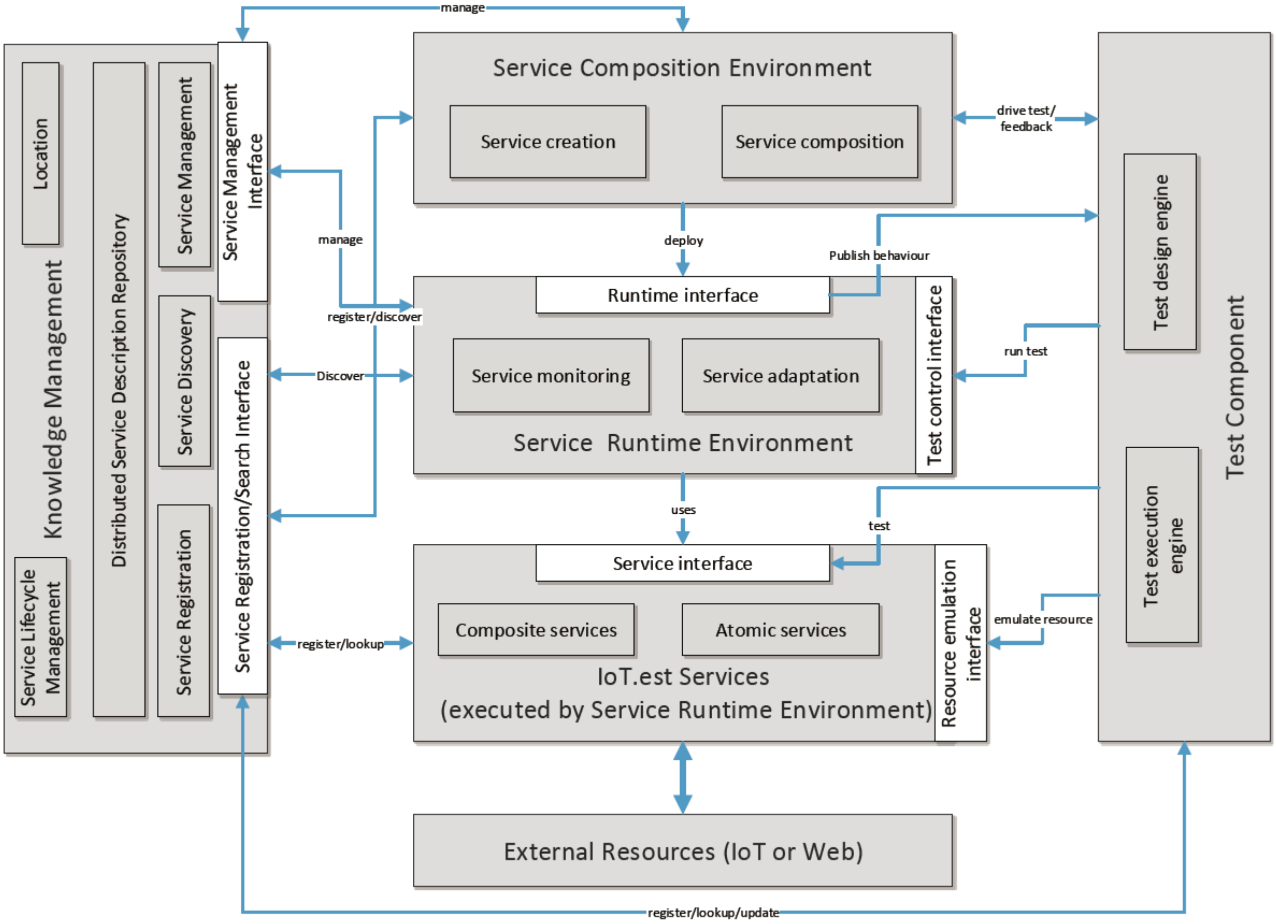


Figure 4 - Overview of the IoT.est architecture

The figure above shows the high-level abstract architecture for IoT service creation and testing. The purposes of the six major components in the architecture are summarised as follows:

Knowledge management: this component is responsible for registration, storage, search and query of the (IoT based) service descriptions as well as some management tasks. IoT.est services will be described and annotated according to the semantic description ontology and knowledge base which can be external to the project (e.g., reusing of the existing ontologies and knowledge base). The semantic service descriptions are stored in distributed service registries. The component also contains a search and query component that is able to locate appropriate services in response to service lookup and discovery queries. It is anticipated that an extraordinarily large number of services will be exposed in the future and these capability constrained services are usually unreliable and often operated in highly dynamic environments, more effective and efficient service discovery and lookup techniques are needed. The component also contains management functionalities related to the service lifecycle and roles.

Service composition environment: this component fills the gap between business goals and IoT processes providing means to transform each of these goals into a composite service in a semi-automated way. These composite services are composed based on atomic services semantic descriptions and the activities that they can perform. A service design graphical user interface is implemented to facilitate the service creation and composition as well as handling of interactions among different components inside the service composition environment.

Service runtime environment: Service runtime environment enables provisioning of IoT enabled business processes. It is related to the deployment and execution phases of the service life-cycle.

Test Component: The derivation and execution of tests of semantically described IoT.est services is managed by the Test Component. The test derivation is triggered by the Service Composition Environment. It fetches the service description from the Registry and search/query engine where it also stores information about its test results. It handles the testing of the IoT.est service in a controllable instance of the Service Runtime (Sandbox Instance) and emulates the external Resources.

Monitoring and Adaptation component: This component is in charge of monitoring the environmental context and network parameters, as well as the correct functioning of the already deployed composite services. Once it discovers a malfunctioning of any of these composite services, it initiates an automated service adaptation process to fulfil service level agreements.

IoT.est services: This component represents the collection of the IoT services and reusable service components developed in the project. Since IoT.est's target is to allow IoT specific services to be described, annotated and bounded in a uniform manner, the term service is generic and not linked to any fixed options regarding the protocol, input/ output format or any other specific SLA details.

External Resources: The external resources are those not designed and developed within the IoT.est project. The resources can be services which can be discovered and used with IoT based services for service composition.

Test enhanced IoT service life cycle framework

The service life cycle addresses IoT specific needs and enables a service-oriented approach to support easy composition and rapid development of IoT applications. Automated test processes supported by the service life cycle are a key technology to achieve these goals. The proposed service life cycle, utilising the IoT.est architecture, enables the following innovations:

- Concepts of service-oriented architectures are transferred to the IoT domain by adding domain specific semantic service descriptions stored in a knowledge management framework. This enables well-known technics for service development and integration in the IoT domain. The knowledge management is accessible by all architecture components as a single point for service descriptions (see Deliverable D2.2). It lowers the obstacle of reusing IoT resources by avoiding the reimplementation of heterogeneous interfaces of silo architectures.
- Early and integrated testing is enabled by a defined service life cycle management, which coordinates the annotation and updating process of the service description between involved components and entities. It enables an automated beginning of the testing process as soon as all necessary information is accessible without manual initiation and early detection of insufficient service description.
- Automated test case creation by analysing basic interface descriptions leads to the creation of test case stubs without the service related parameterisation. It requires full manual provision of valid and invalid test data. In the IoT.est service life cycle semantic models provided by the knowledge management are reused to enable semi-automatic test derivation by extensively describing service in- and output parameters as well as the service behaviour. By utilising service description knowledge it is possible to create abstract test cases. By combining this information with domain knowledge for physical value interpretation the creation of reasonable test cases can be achieved by constrained value ranges for the specified data types (see Deliverables D2.3 and D4.1).
- Previously the involvement of various roles in the test process and frequent implicit information hindered a rapid development process of IoT services. In IoT.est explicit and machine interpretable knowledge representation, which is semi-automatically collected during previous service life cycle stages, enables a deterministic test execution. The test execution therefore utilises the service descriptions for selection, configuration and control of the sandbox environment during deployment and testing (see Deliverables D2.3 and D5.1).

IoT.est distinguishes four service life cycle phases belonging either to design- or run-time. Design-time is understood as the service creation time that splits into modelling and composition phases. Run-time aspects cover service provision that can be subdivided into service deployment and service execution. In IoT.est all phases are characterized by a knowledge based, test driven approach.

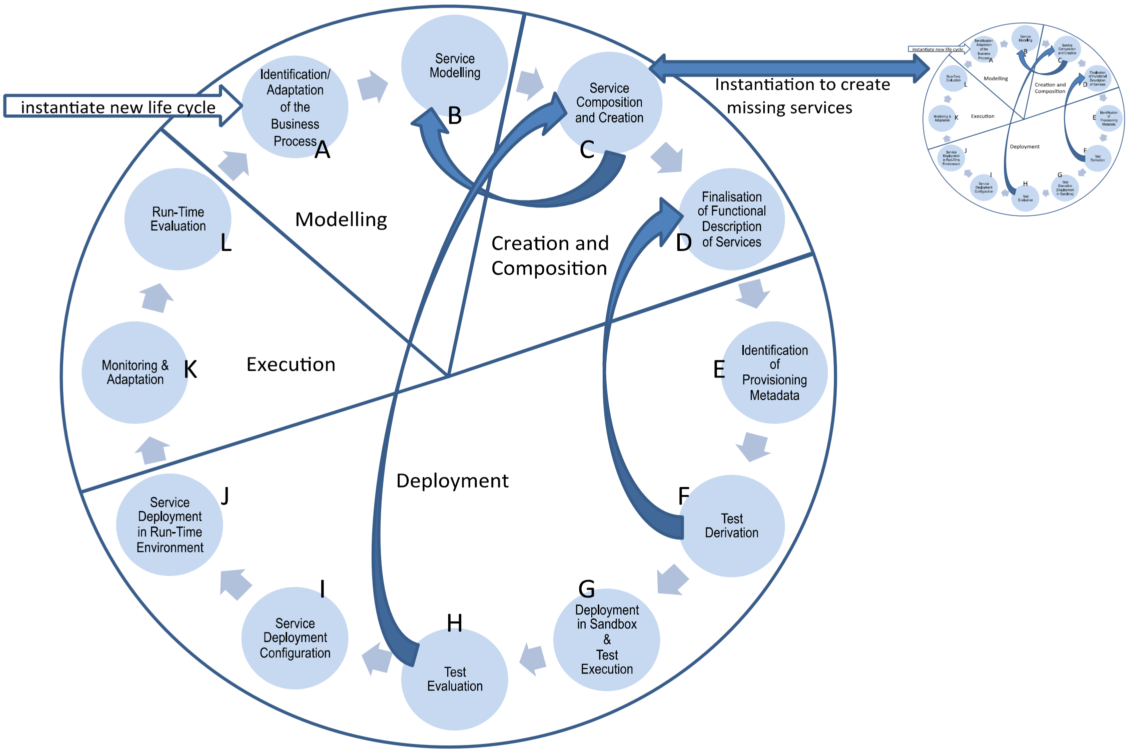


Figure 5 – Detailed Life Cycle Description

The Service Modelling phase consists of two stages: identification/adaptation of the business process and modelling (the IoT.est project utilises a semantic approach for service modelling). The identification/adaptation of the business process is an iterative procedure to derive business use cases, requirements and specifications, which are then translated to technical specifications and formal knowledge representation for services in the modelling stage.

The Service Creation and Composition phase represents the developer related phase in the service life cycle. The main intention is to reuse available services and to create/develop missing ones. The finalisation of this development process is reached with the full-qualified functional description of service in-/output parameters and behaviour.

The Service Test and Deployment phase describes the deployment process including the necessary test process, which is eminent before the usage of the service in a productive environment. After the identification of the provisioning metadata, which is used to identify the runtime capabilities necessary for the test deployment, sandbox-specific test cases can be generated. After the deployment in the sandbox environment and the test execution an evaluation stage is scheduled. The Service Deployment can be used in two different situations:

1. The service is deployed in the testing sandbox runtime environment for testing purposes
2. The service has already been tested and certified as ready to be launched in the marketplace.

In this case the service is deployed in the production runtime environment The Service Deployment Status model described below is applicable for both situations and only the runtime context will be different (e.g., Runtime Platforms used and the Service Provisioning Metadata).

During service execution the Runtime Platform management services, monitors services' status against QoS described in the Service SLA and if needed an adaptation process is performed. Each of these major Service Execution stage processes has their own state machine

Contributions to the State-of-the-Art:

- Proposed architecture for the future business of Things, published at [8] (see Section 1.4.1)
- Proposed Business of Things architecture, published at [16] (see Section 1.4.1)

2.2 Semantic Modelling and large scale IoT service discovery

The semantics-based methods and tools developed within the IoT.est project have been designed to support the modelling, creation, composition and testing stages in the service life cycle. The IoT community proposes to use semantic technologies for describing and annotating IoT resources and entities of interest. This has the potential to enable representing, storing, interconnecting, searching and organising information related to or generated by heterogeneous things. Together with semantic technologies, service-oriented principles can be applied to the IoT, which can facilitate the development of large-scale, loosely-coupled IoT based applications and services. Thus, the IoT.est project's approach is to abstract IoT resource functionalities and capabilities in terms of standard service interfaces to support uniform service operations.

The Knowledge Management (KM) component, shown in figure 6 below, encapsulates the key functionalities of semantic annotation for IoT services, service description distribution/storage, and IoT service lookup and discovery, which are the building blocks of the project and are essential for other work packages in service creation, composition, testing and deployment.

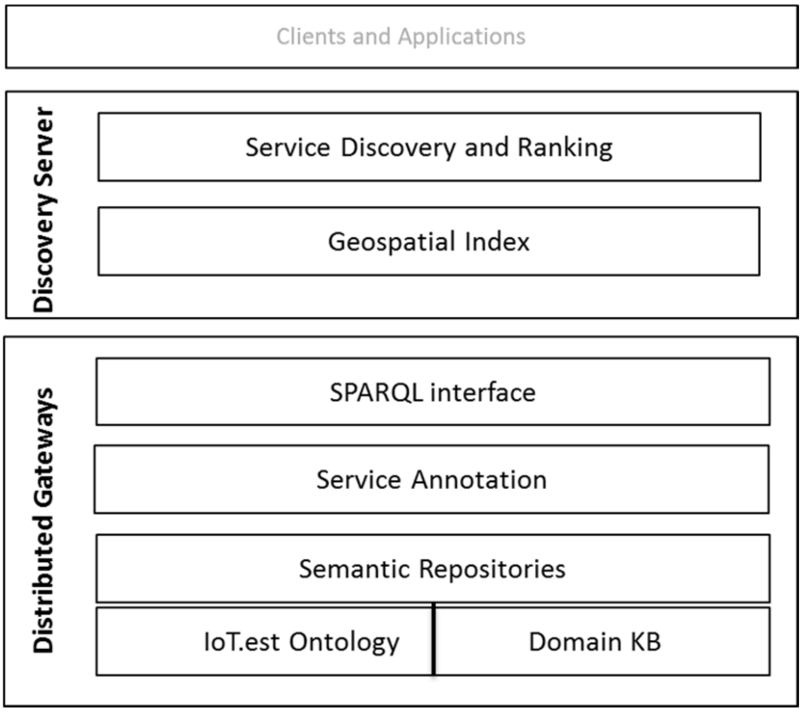


Figure 6 – Knowledge Management Component

The key parts of the KM and their functionalities are as below:

1. Gateway: in the KM framework, each service is associated with a gateway, which provides resources to host the services and includes a semantic repository to store the service descriptions. The semantic repository acts as a RDF database (or triple store) and is implemented using Sesame. The IoT.est ontology underlies the semantic repository and service annotation and provides the required formalism and common, structured model for the other sub-components of the KM.

    i. *IoT.est description ontology*: The ontology modules provide the required formalism and common, structured model for the other sub-components of the KM. The developed suite of ontology modules represent IoT services the IoT resources underlying these services, service resource requirements and platform capabilities (in which the service could be deployed), as shown in the figure below.

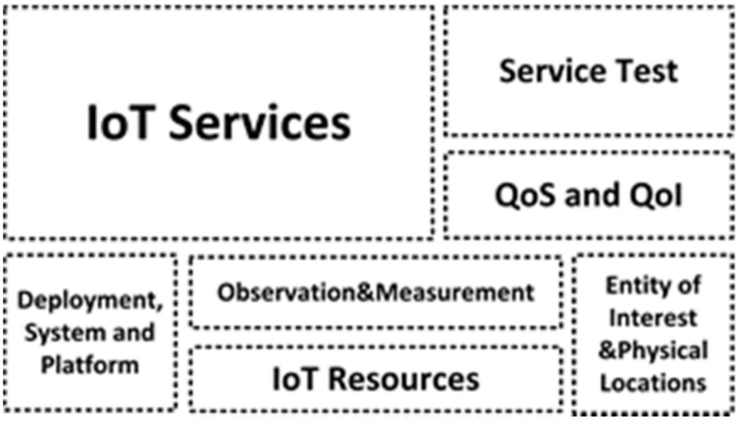


Figure 7 – Modules in the IoT.est description ontology

Modelling methods on the IoT Resources, Entity of Interest and Physical Locations, and Observation and Measurement have derived by reusing existing works, such as the SSN ontology and the IoT-A entity-resource ontology.

The service aspects captured include the profile-method-grounding aspects. The test module concepts are used for verifying functional and non-functional capabilities of IoT services during design and deployment stages. The Platform ontology captures capabilities such as hardware, software as well as the networking capabilities of the deployment platform. The semantic annotation of the platform capabilities enable run-time automated deployment configuration of the composite service by matching the capabilities of the deployment platforms to the requirements of the IoT services.

Services stored in the semantic repository are annotated according to the IoT.est description ontology by using the developed Service Annotation Tool (SAT). The SAT provides a simple Web-based form interface for a service developer to supply values for information fields according to the IoT.est service ontology. The SAT creates a service instance using the supplied values that are automatically mapped to the relevant fields in the service instance. The SAT also performs automated parsing of the service WADL files to generate the grounding aspects of the service annotation and enables the service developer to link these with IoT resource operations.

2. Discovery Server: In the discovery server, the geospatial index helps the discovery engine reduce the search space and locate the gateway(s) that are likely to contain the services with respect to the queries.

A SPARQL search is then performed on the repository to search based on other specified query parameters.

The gateways forward the aggregated geographical information of all the stored sensor services to the indexing server, which uses the R-Tree indexing method to enable efficient search. The bounding rectangle of the gateway encloses the observation areas of all the sensor services registered to that gateway. With the geographical bounding rectangles, gateways become spatial objects and therefore can be indexed using the R-Tree. In the IoT.est project, the geospatial index of R-Tree indexes the gateways based on their minimal bounding boxes, instead of indexing individual IoT services. Such design can not only support efficient search based on geospatial indexing, but also can limit the number of update operations which are frequently needed in IoT environments. Update operations in R-Trees can be computationally expensive; however, in the proposed approach, by indexing the gateway components (not the individual IoT service), most of the changes on services are constrained within the managing gateways. Update messages do not need to propagate to the higher levels in the R-Tree, therefore, this significantly reduces the computational complexity of the update operations in the dynamic environments.

The KM provides the following functions to the other components of the IoT.est architecture:

- **Service discovery and ranking**: the KM allows for queries that can specify a number of search criteria, such as service and/or IoT resource location, service category, related entity, input/output parameter types etc. in addition to the SPARQL semantic search, approximation functions are supported for the location parameter, wherein both geo-coordinate and logical location search support ‘a degree of match’. For geo-location, users can specify that the search be performed within a certain distance of a given spatial point. For logical location, the semantic search takes advantage of the containment relations asserted in the indoor location ontology to offer different search criteria, e.g. in the same room/floor/building. Weights attached to different search parameters also enable the search results to be ranked, with an overall matching score computed for each retrieved matching service.

- **Platform recommendation**: this functionality supports deployment configuration by recommending the most suitable platform to deploy a composite service. The recommendation engine retrieves the corresponding semantic description of the service to get the associated QoS parameters, deployment description as well as requirements of the service executable along with the semantic descriptions of the available deployment platforms. These parameters then serve as requirements to the reasoning engine to determine which platform can best host the service.

- **Service adaptation recommendation**: when key SLAs for a composite service are broken, the monitoring component can invoke the KM to search for suitable replacement services. The recommendation engine accepts the service ID of the failed atomic service and its endpoint URL and retrieves its semantic description that acts as a template for finding matching alternative services. The IOPE signature of the failed atomic service is taken as a mandatory parameter to search for alternative services, with additional parameters such as category, location etc. acting as optional flags with pre-defined weights according to the replacement policy. The list of retrieved alternate services is ranked according to the weighting criteria and returned to the Adaptation Manager, which routes the malfunctioning atomic service endpoint to the top-ranked alternative service instance.

Contributions to the State-of-the-Art:

- Proposed ontology for IoT, published at [1] (see Section 1.4.1)

- Semantics for IoT, published at [5] (see Section 1.4.1)
- Semantic sensor service networks, published at [6] (see Section 1.4.1)
- Computing perception, published at [7] (see Section 1.4.1)
- Semantic matchmaker for IoT, published at [10] (see Section 1.4.1)
- Semantic modelling, published at [12] (see Section 1.4.1)
- Distributed hybrid matchmaker, published at [13] (see Section 1.4.1)
- Information abstraction, published at [21] (see Section 1.4.1)
- Geospatial indexing, published at [24] (see Section 1.4.1)

2.3 Goal-oriented IoT Service Composition and Testing (Design-Time)

One of the main goals of the project is the development of a goal-oriented and knowledge-based service composition environment (SCE). This SCE also integrates testing capabilities for both atomic and composite services as well as the main results of the technical work like a ranked searching request based on the semantic annotations of each service and automated deployment and execution of composite services. This SCE aims to bridge the gap between various business services and the heterogeneity of networked sensors, actuators and objects. The SCE focuses on the Internet of Things domain with a large number of resources and the need of automated interpretation of environmental and context information.

Although the ideal functionality of the SCE will be to fully automate service composition, it is not always possible as there is a probability of not finding an atomic service that fully fulfils user's needs or expectations or even there is no service with such functionalities. These situations are shown in the following figure:

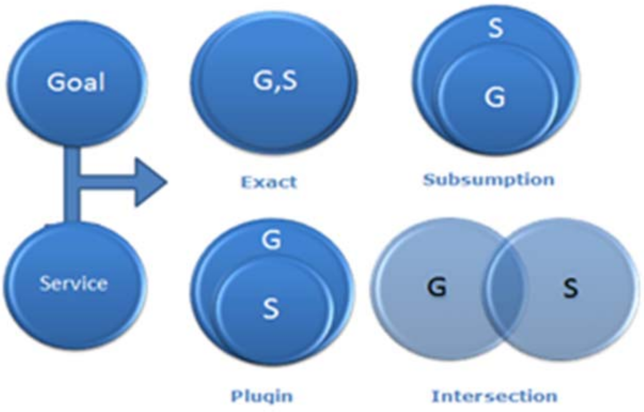


Figure 8 – Possible search results

The SCE focuses on the composition of IoT services in a semi-automated and efficient way based on the business process goal in an application domain. The SCE allows the identification and validation of re-usable service components and process flows that defines the composed services and their functionality before deploying them on the IoT platforms.

The end user has an important role in the SCE, because it drives the composition and also handles the start of the testing and the deployment. All these functionalities are exposed through a GUI in the front end, but on the other hand it also orchestrates the request to some of the internal components on the back end. The main actions covered by this GUI are:

- *Goal oriented and knowledge-based IoT service composition:* adding reasoning on top of the semantic search in order to assist the end user to find the best matching service.
- *Test design:* in order to start test derivation for getting all the possible test cases per service and then executing them.
- *Deployment and execution:* in order to make deployable and executable a composite service, it is represented using standard BPM2.0 in a simple file that contains all the information related to the composition.

The following diagram depicts the main components of the SCE and their interaction:

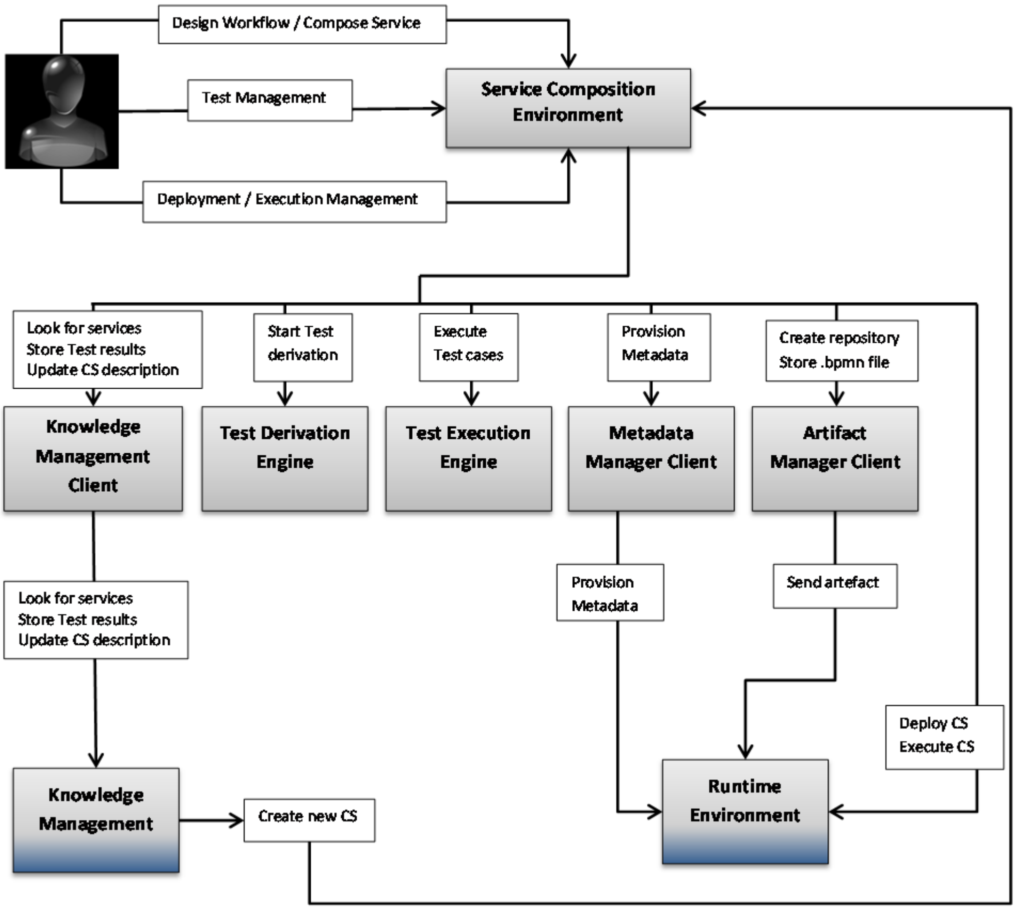


Figure 9 - Main Components of the SCE

The components represented in the diagram above can be grouped into four main ones. They are described below in terms of the functionalities that they provide, their limitations, dependences and technologies used.

1. **Service Composition Environment**
   The Service Composition Environment component has the same name as the whole SCE developed mentioned before. It contains the key features for designing goal-oriented business workflows and creating service compositions. These actions are:
      - Design of a business goal (workflow).

- Create a service composition. This is a generic action that involves different actions at the same time:
  - Search for services given specific criteria.
  - Trigger atomic services testing and collect results.
  - Bind services into the composition.
  - Update the annotations of a composite service.
  - Provisioning Metadata and store the resultant .bpmn file.
  - Trigger test derivation for composite services and collect results.
  - Update the Knowledge Management with test results.
  - Deploy composite services in the Runtime Environment.
  - Execute composite services in the Runtime Environment.

The user can create workflows, search and discover services, bind services to service tasks, test, deploy and execute them by using this platform. The SCE component provides an interface, showing the requested information in each case and analysing the results.

2. **Knowledge Management**

The Knowledge Management component provides a number of functionalities to the SCE through defined interfaces to achieve a goal-driven service composition. Crucially, it assists the designer during the service composition by providing reasoning support for matching services to the different workflow blocks. Semantic matchmaking and ranking is employed to match services to each workflow item.

The knowledge Management functionalities relevant to the SCE include:

- *Creation of composite services.* This is done through the Service Annotation Tool (SAT), that presents a simple web form that user can fill with the relevant information. Once the form is submitted, a semantic search service description for that service is automatically created and the Knowledge Management component will store this service description in a triple store.
- *Discovery of the atomic services that could be candidates for composition*. When the SCE needs to create the composition, first it calls the discovery service in order to search for (discover) atomic services that can match the different tasks of the designed workflow.
- *Semantic annotation of composite services*. This is done through the Annotation service that updates the KM with the composite service fields once the composition process is complete.

3. **Test Component**

The Test Component is also integrated within the SCE, allowing the user to test both atomic and composite services before using or deploying them. For more information about this component, see Section 2.5

4. **Runtime Platform**

The SCE interacts with the Runtime Environment (RTE) through the Metadata Manager client, to save the service provisioning metadata, and with the Artefact Manager client, to store all service files needed, notably the .bpmn file containing the composition workflow logic.

- *Metadata Manager client:* The Metadata Manager client interacts with the RTE Metadata Manager to provide data needed to provide the service the most suitable runtime platform available. The Service Provisioning Metadata provided by the Metadata Manager client includes information about the service implementation, the interfaces that can be used to access to it and the service components comprising the structure of a composite service.

- *Artefact Manager client:* The Artefact Manager client interacts with the Artefact Manager to create the Service Artefact Repository and store the .bpmn file in the created repository.

The SCE performs the majority of the operations within the whole environment, acting as the core of the system, but without exposing any interface as it is the one consuming them. This means that all operations are centralized on the SCE, and it takes care of calling the appropriate method of each component to perform any action, from searching services to executing a composite service.

For this reason, the SCE offers a GUI to consolidate the usage of all involved components, allowing human interaction when it is needed or just showing results on the screen. As most of the operations are performed on the background, this GUI allows the user to monitor the progress of the operations that is performing.

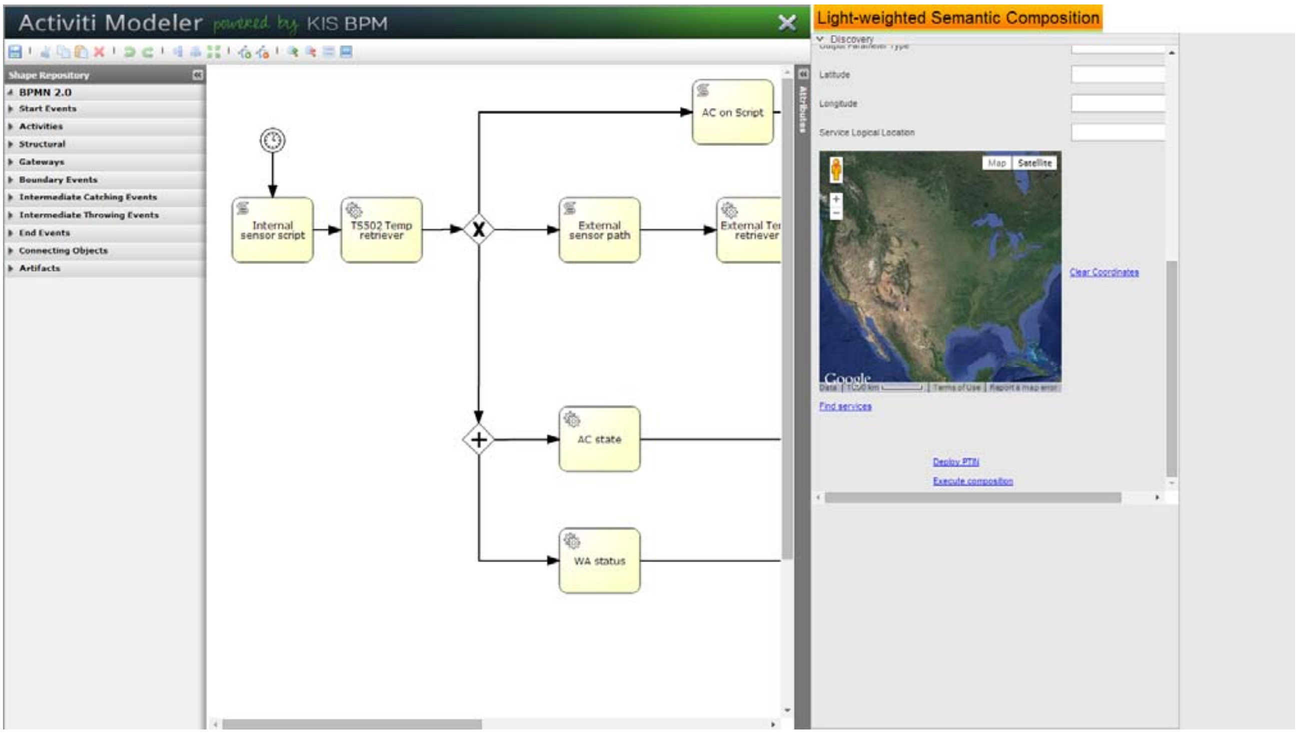


Figure 10 - SCE GUI

As it can be shown in the above figure, the GUI is divided into 3 panels:

- *Left Panel:* it contains the list of all elements, supported by standard BPMN2.0, which can be added to the workflow design.
- *Central Panel:* it contains the graphical representation of the business workflow. In order to ease its design, all the elements of the left panel can be dragged and dropped to this central panel and they are automatically represented.
- *Right Panel:* this panel is called Light-weighted Semantic Composition, as it is the one which offers the functionalities for creating a service composition.

The main novelties of the Service Composition Environment are:

- The introduction of the goal-driven and knowledge-based service composition, based on semantic descriptions in order to determine the suitability of the services.
- The inclusion of reusable services for IoT business modelling as they are agnostic of the domain and can be incorporated in different compositions avoiding the duplication of services with the same functionalities.
- An innovative approach for automated test derivation, further explained in section 1.2.5.

These functionalities end in a Service Composition Environment fully operable, which incorporates all the needed tools to cover the IoT.est service lifecycle.

2.4 Automated IoT Service Provisioning (Run-Time)

The ubiquitous spread of smart devices with communication capabilities will make the development of intelligent services orchestrated by business oriented processes possible, running over a world-wide large scale distributed and heterogeneous environment. IoT services' life cycle is deeply affected by the dynamics of the physical world. Physical and virtual objects may have very volatile characteristics e.g. they can move along different physical places and they may have very short and intermittent active life-cycle stages. Such volatility can introduce severe errors in IoT services execution forcing IoT services to adapt to different contextual environments in order to ensure the service provisioning with the agreed quality. WP5 IoT.est work addressed such complex problem by using a pragmatic combination, in a common and consistent IoT runtime framework, of emerging Service Oriented Computing concepts, test-oriented resource emulation models and rich semantic concepts.

A new **Distributed Runtime Environment** model (see Figure 11) and architecture is presented based on emerging industrial standards which are complemented with semantically annotated models enabling semantic matching between service requirements and available runtime platforms for Quality of Service (QoS) aware provisioning. Such model promotes the re-usage and extension of existing SOA runtime solutions with no need for new runtime solutions built from scratch thus increasing the chances of having IoT.est concepts adopted by the industry. The IoT.est RTE model is composed by different Runtime Platforms where services are deployed and executed. Each Platform can support different deployment and runtime technologies and it can be hosted either in powerful computing environments built on top of cloud computing technologies - Cloud Runtime Platform - or it can be hosted in much more computing constrained device environments - Device Runtime Environments. At deploy time Composite Services could be distributed over different Runtime Platforms to better cope with required Composite QoS. Already deployed Component Services can be reused by the Composite Service in case they don't potentially introduce non-compensable disturbances into the Composite QoS.

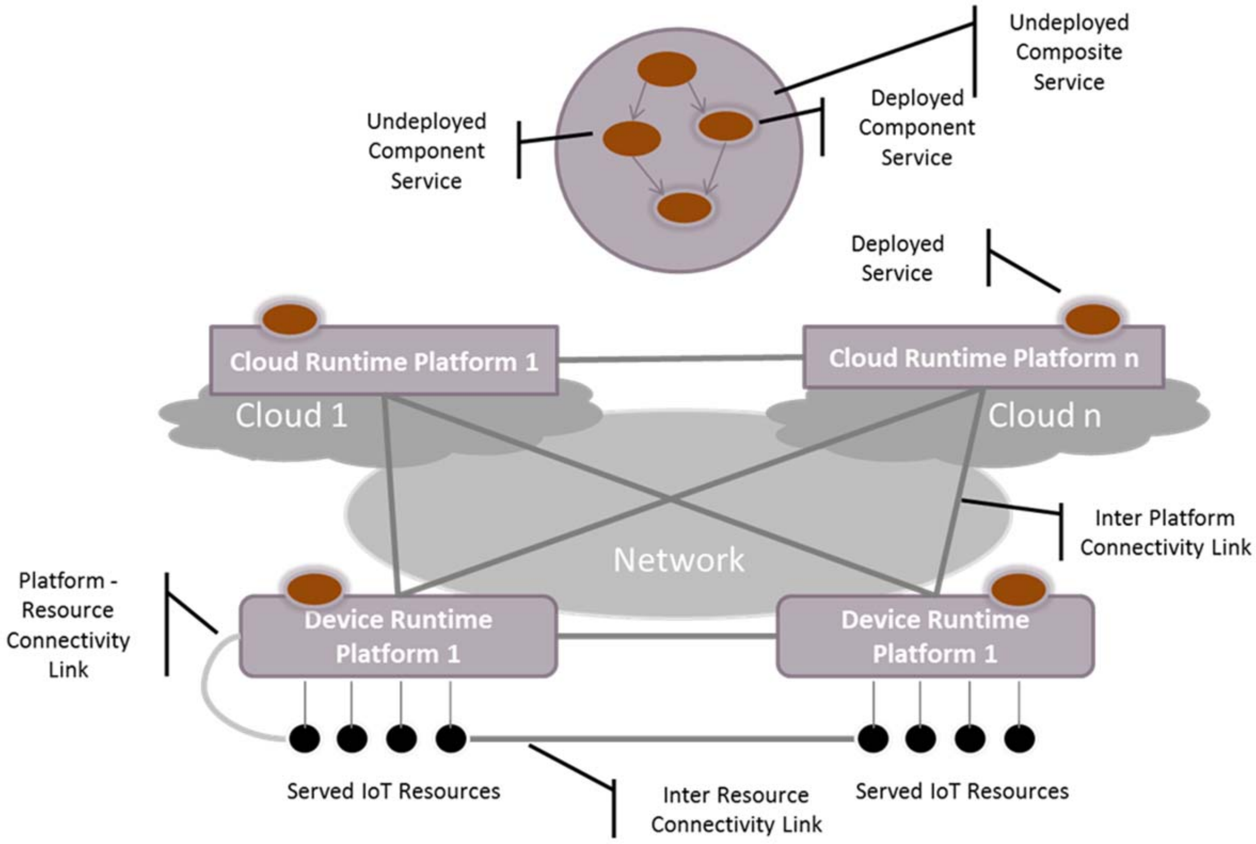


Figure 11 - IoT Distributed Runtime Environment Model

The Runtime Platform Served IoT Resources is a major new concept introduced by this model. It defines the IoT Resources that are accessible from the Platform in privileged conditions. These means IoT resources should have local network connectivity to the runtime platform e.g. the Runtime is hosted in a Gateway that directly interface with sensor networks like a local Bluetooth or a Zigbee network. Otherwise, Serving IoT Runtime Platforms should have remote but good network connections to served IoT Resources e.g. gateways are connected via broadband access networks to Data Centers where the runtime platform is hosted and both are managed by the same Telco operator. Served Resources must be encapsulated by IoT Atomic services running in the serving platform. For availability purposes, one IoT Resource should be served at least by two distinct runtime platforms. Typically, Cloud Runtime Platforms only support remote network connections to Served IoT Resources and Device Runtime Platforms support local network connections to Served IoT Resources.

The main RTE points of novelty are:

- A new semantic model is developed to support automated service deployment during runtime. Moreover, it enables automated runtime adaptation decisions when a service faces QoS problems.
- A new overall approach of Sandbox testing of IoT-based service is specified and implemented. The systematic approach enables enhanced capabilities for automated testing of IoT-based services before deployment in the life environment.
- A new distributed runtime environment model and architecture is proposed to handle major scalability and dynamicity challenges of the Internet of Things. It provides the foundations to realistically enable dynamic QoS-aware provisioning of composite IoT services in world-wide scale runtime environments.

2.5 Monitoring and Adaptation

The **Monitoring and Adaptation** component support service level monitoring as well as composite service adaptation. The former functionality refers to the ability of the platform to monitor atomic service KPIs (Key Performance Indicators) which can be used to determine the service performance or other relevant parameters for its operation.

The latter functionality deals with the reconfiguration of a composite service if one or more of its atomic services fail to meet the predefined limits set for the KPIs they are exposing. Adaptation refers to the atomic service replacement inside a composite service.

Monitoring of an atomic service starts whenever that atomic service is part of a composite service which has been deployed and is executed. Once a composite service is deployed, all of its atomic services are requested to register the KPIs they are exposing. This is performed using a dedicated service interface which the atomic services should expose. Once registered, KPIs are periodically queried and compare to their corresponding threshold values. Whenever a KPI fails to meet its requirements the adaptation process is triggered.

Service adaptation involves two steps. The first one is the retrieval of an suitable atomic service which could replace the one with failed KPIs. The second step involves the actual replacement inside the composite service.

Alternate service retrieval is performed based on the semantic description of the failed atomic service. The first search criteria relates the interface compatibility (data types, parameter format, semantic type of the parameters). This is a mandatory requirement and is application independent. Other criteria are application dependent and may refer to geographical location, logical location, referred entity, etc. All these fields are part of the atomic service semantic description and can be used to define a proper replacing service.

Current version of the adaptation components only takes into consideration a limited number of fields but its extension can be done by attaching replacing rules to each atomic service.

Searching a replacement for the services is performed by the KM once the adaptation process is triggered. Once found, the best replacement candidate is used and the composite service is redeployed by the RTE.

Foreseen developments of the monitoring and adaptation components are related to the KPI evaluation as well as to the service replacement criteria, where rule based solutions can be applied in order to support application specific requirements.

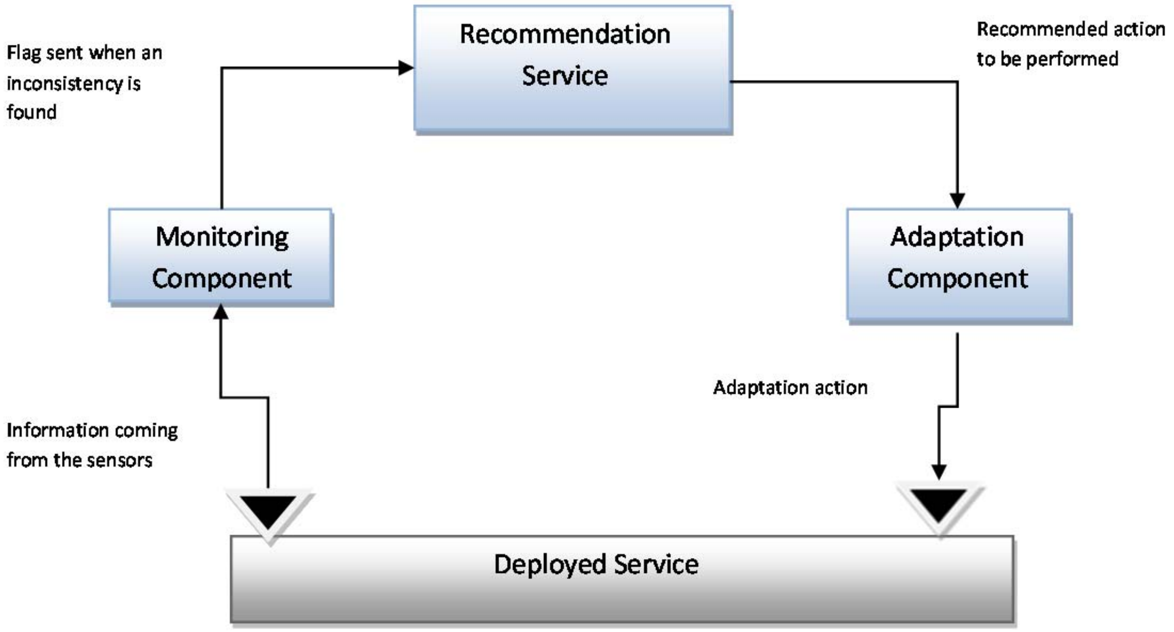


Figure 12 - General Overview of the Monitoring and Adaptation component

The main points of novelties of this component are:

- An innovative approach based on 3-layer monitoring is introduced to allow IoT services to adjust to the current environmental changes.
- Automatic adaptation is included in order to re-do the service composition substituting the atomic service that is causing the malfunctioning.
- A recommendation service that is able the most suitable service to be substituted, based on its semantical annotations.

2.6 Test Component

The Test Component is responsible for the derivation and execution of tests of semantically described IoT.est Services. Figure 13 shows the updated interaction with the main IoT.est architecture components together with the identified internal functional test components.

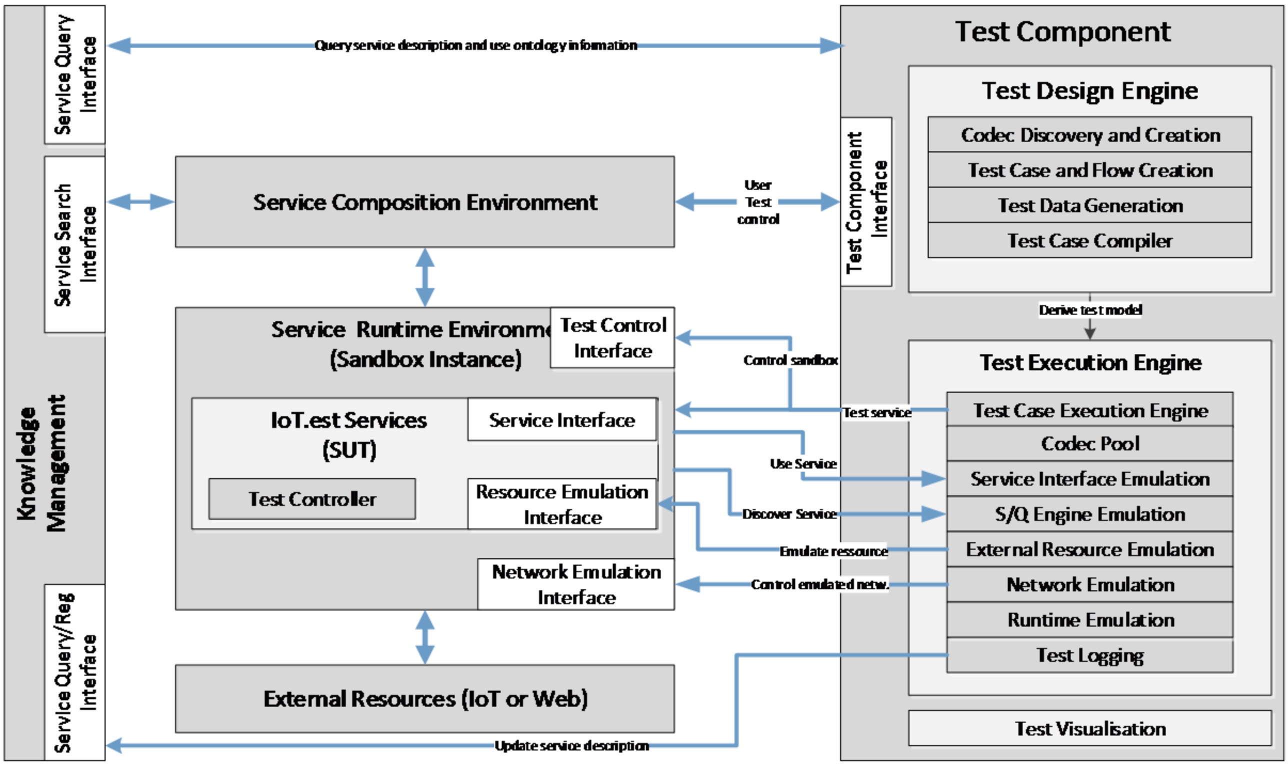


Figure 13 - Test Component Architecture

The Test Component has two main components, the Test Derivation Engine (TDE) in charge of creating test campaigns that contain all needed test cases for both atomic and composite services, and the Test Execution Engine (TEE), which executes the test cases given for a specific service and returns the results of the functional conformance tests to the SCE so it can update the Knowledge Management with the overall results.

Test Derivation Engine: The derivation is driven by processing service descriptions and utilising domain knowledge. The descriptions stored in the Knowledge Management contain the information used to find matching services for compositions and derive functional tests. To obtain a comprehensible test generation, the TDE utilises an explicit information representation approach, which can also be used to evaluate and alter the model and tests, which are automatically derived.

Test Execution Engine: The SCE interacts with the TEE just after finishing the test derivation step. This component is responsible of executing the test cases that are defined in the test campaign (.clf file) provided by the TDE. After all the tests from the given test campaign were executed, the SCE stores the results of the execution process in the Knowledge Management.

The *Sandbox Testing* approach is responsible to enable the execution of different types of test cases (unit, integration, regression, non-functional tests) within a controllable and realistic behaviour of the surrounding environment. While the semantic models describing the IoT-based services, the platform and the network resources are utilised to derive the test cases, thus enabling the Sandbox to execute these test cases.

The functional and non-functional behaviour of IoT-based services is mainly influenced by three different environment influence categories:

- The Network resources (e.g., delay, packet loss),
- the Service Runtime Environment (e.g., load of the platform),
- and the connection with IoT/ external resources and their interaction with the real world.

The interaction with these different categories influences the behaviour of the SUT. From the test perspective the relation can be described as there is a stimulus that comes from the environment and this stimulus might result in an effect from the SUT. The aim of the Sandbox testing concept is to identify if some parameters or a combination of parameters can result in wrong or unexpected service behaviour. The concept is based on fully controllable emulation components which enable the fast and safe execution of the SUT in a Sandbox environment.

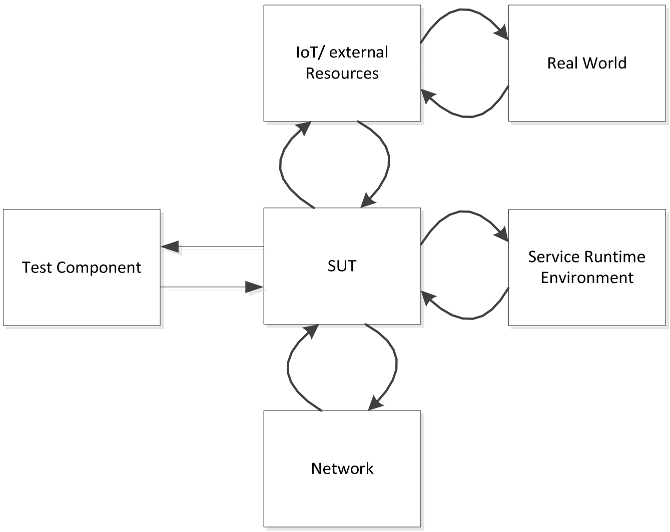


Figure 14 - System Under Test Influence Behaviour Model

IoT resources consist mainly of sensors and actuators deployed sometimes together depending on the application requirements. Actuation, be it opening of a window or turning on a heater, is most of the times accomplished in conjunction with a feedback loop which is able to provide state information related to the actuation process (for instance, the door has been opened or the temperature in the room reached a certain threshold value).

Still, some applications do not require any form of feedback as there is no actuation. For instance, basic sensor reads such as those of weather sensors or fixed surveillance cameras can be accessed by multiple clients without any problems. When actuation is involved, in an IoT environment an important aspect which has to be considered is the fact that actuator services can be accessed by multiple clients thus leading to possible concurrency problems. Real-life testing of such scenarios is thus much more difficult to be achieved compared to the case when just “read-only” services are involved.

In order to overcome this problem a possible solution is the emulation of these services whose goal is to mimic the behaviour of the underlying *thing.* While this kind of emulation requires additional effort and sometimes is not able to reproduce completely the behaviour of the real *thing,* it comes nevertheless with a couple of advantages:

- testing can be performed without the risk of interfering in an inappropriate manner with the real IoT infrastructure or the environment where this is deployed;
- tests can be performed at any time as there is no constraint about the availability of the infrastructure;
- the emulated resources or the environment where they are “deployed” can be changed at any time, increasing the number of possible testing scenarios;
- error injection (at different levels – device, data format, communication) for testing purposes is much easier to be implemented;
- Increased scalability as the number of emulated resources can be tuned according to the test case needs.

An example of resource emulation is provided in the *Energy Efficient Buildings* scenario. There, temperature sensors, heating, ventilation and air conditioning (HVAC) controls, windows and windows blind actuators are deployed in a simulated office building scenario. These resources are emulated and expose their data as IoT services. Besides the normal behaviour, they also provide the means of introducing faulty behaviours, which have to be taken into consideration during the development and testing of the IoT application.

The main target of the test execution approach is i) to have an overall methodology to describe and include the emulation of different influence factors and ii) enable a dynamic emulation of these factors, based on the current test case. It is part of the on-going work to identify if and how dynamic adaptation can help to reduce the effort of testing. Similarity based test cases selection is a promising approach to enable test space diversity with an adequate failure detection rate and thus can reduce the number of required test cases [Hemmati10]. The minimizing of the required test data space is a NP-hard problem (set cover optimization) and therefore exhaustive search approaches are not feasible. Instead, minimization algorithm either based on clusters (although clusters are nor minimizing algorithm per se) or search based approaches (random, hill climbing, evolution algorithm, etc.) with a stopping criteria are candidates to minimize the number of test cases. It is part of the future work to evaluate if these minimizing techniques can be utilised in the IoT domain and are adequate to identify important use cases, which also takes the behaviour of the environment into account.

The followed concept is also driven by the assumption that the process of selection of test data can also be a dynamic process. We propose that after the first set of test cases with a good diversity (based on the outlined search or cluster based approaches) is executed the test results needs to be evaluated and can be utilised to concentrate on most interesting parts of the test data space. One possibility would be to identify the most critical test results (compared to the description of the service e.g. QoS) and use this as a starting point for new test data generation and selection based on a minimizing algorithm (e.g. data mutation around critical values).

Different tools, metrics and patterns (e.g. Rules) will be designed and developed, according to the specific IoT.est needs that will be used to reason about the monitoring results in order to determine whether to perform an adaptation action or not.

Contributions to the State-of-the-Art:

- Novel concept for IoT service testing based on model-based testing, published in [2] (see Section 1.4.1)
- Concepts for test automation in the domain of IoT, published in [11] (see Section 1.4.1)
- Concepts for sandbox testing of IoT services, published in [14] (see Section 1.4.1)
- Novel approach for automatically derivation of emulation interfaces for advanced testing based on IOPE descriptions and rules, published in [15] (see Section 1.4.1)
- Novel approach and initial results for derivation of a System Under Test model out of semantically enriched interface descriptions of atomic services, published in [18] (see Section 1.4.1)
- Integration of model-based test aspects into the IoT service lifecycle, published in [19] (see Section 1.4.1)
- Novel interpretation of algorithms for test case diversity analysis, published in [22] (see Section 1.4.1)
- Proof of Concept for automated model derivation, test case creation and execution for IoT-based RESTful services, published in [23] (see Section 1.4.1)

2.7 Healthcare Showcase

One of the challenges which the project team faced was the integration of all components into a functional solution. This goal was important since complete integration was critical for the demonstrating the proof of concept.

In the early phase of the project a number of candidate scenarios have been created to support the proof of concept activities. Of these, two were selected and later extended in order to demonstrate added features for each of the project components.

The first scenario was the Wellbeing Scenario, where the main character, Maria, is put into different situations in which she interacts with *things* capable of sensing and actuation. Maria lives in an IoT enabled environment, and benefits from the latest technology in the field, Maria's health condition requires constant care and is supported by the applications grouped into the Wellbeing category.

This scenario takes advantage of WebRTC technology that enables Real-Time Communications (like videoconferencing) to run in any web browser without the need to develop and install any plug-in. In addition WebRTC also support real-time communication of data notably data captured by sensors. The Wellbeing scenario leverages these WebRTC capabilities to enable conversations (Emergency Conversation, see Figure 15) among citizens in emergency situations, human assistants (including 112 agents and doctors), devices (sensors and actuators) and processing data services. The Emergency Conversation is able to share with human assistants and processing services data captured by wearable devices like smart watches to provide additional contextual data to assistants. Shared Sensed data is then processed by Data services also involved in the conversation as "non-human" participant to infer more contextual data that can take advantage of other sources of sensed data including location, temperature, historical data from the patient or data collected from other medical equipment to achieve more accurate diagnosis. Then, the emergency situation orchestration uses the inferred diagnosis to automatically perform actions, for example, to add new expert participants (doctors) into the Emergency Conversation.

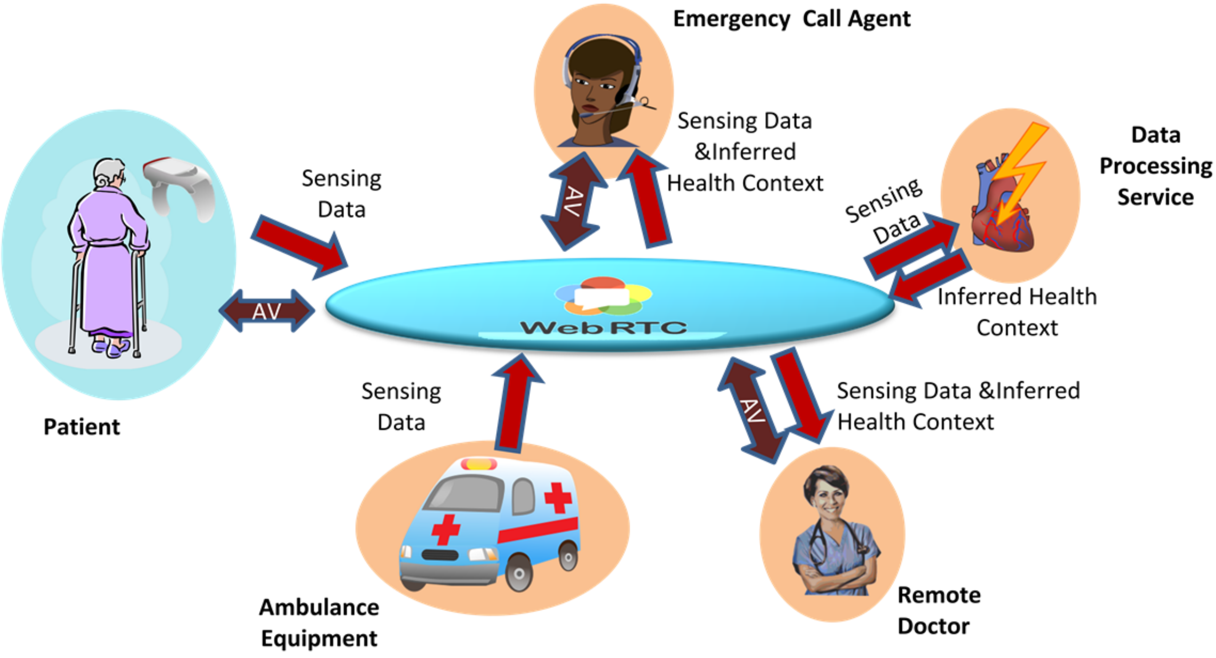
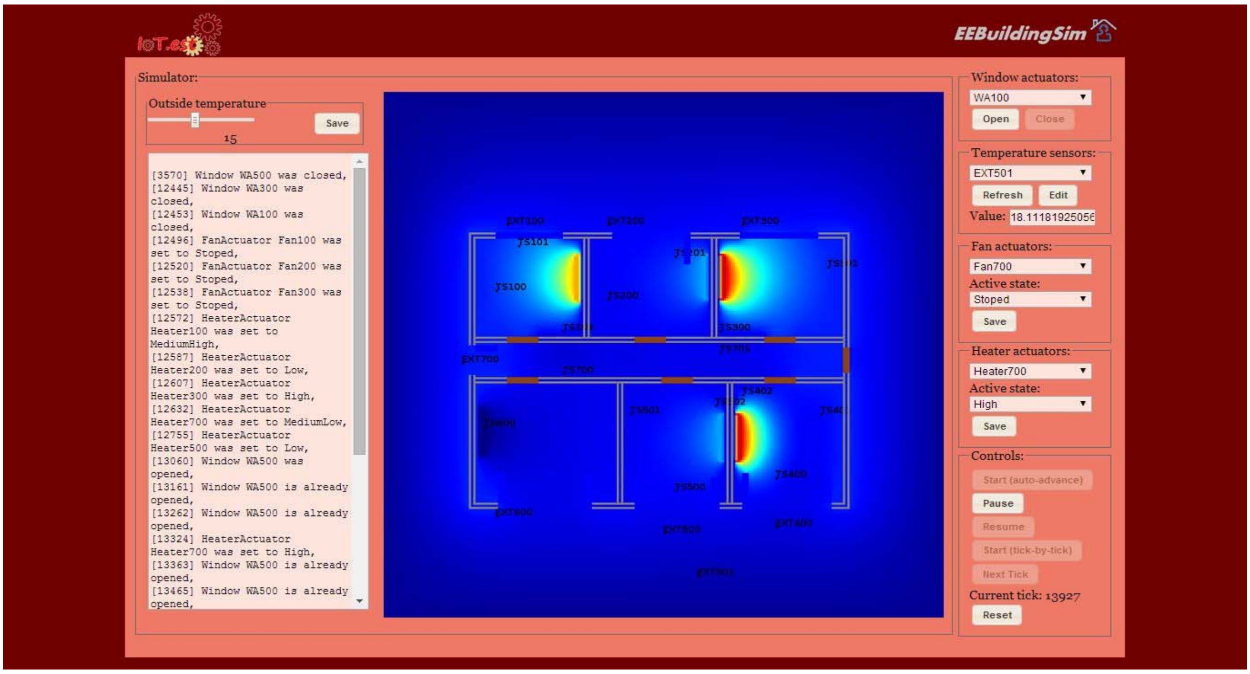


Figure 15 – Wellbeing WebRTC Emergency Conversation

The usage of WebRTC Data Channel and the Protocol on-the-fly concept (FP7 Open Lab Deliverable D4.15, WONDER Assessment Report] provides a new IoT P2P communication paradigm to enable seamless interoperability between IoT services. Such concepts will be further explored in the reTHINK H2020 (ICT 5 – 2014 call) project expected to start in January 2015.

The second scenario, called Energy Efficient Building is focused on the integration of various sensors and actuators deployed into and around a smart building in order to reduce its carbon footprint.

While the former scenario involves the use of real devices (such as a weight scale or a blood pressure monitor) the former one is based on a simulated environment. Having some real *things* which were included into composite services proved that real life use is feasible and allowed us to have a lively demonstrator. On the other hand, the ability to create and deploy as many sensors and actuators into the EEBuildingSim simulator with no costs for physical devices and minimum integration effort, proved useful every time when demonstrating e project functionality required some extension of the scenarios and changes made to the environment. The front end of the simulator is shown in figure 13 below.



Outside temperature: 15

```
[3570] Window WA500 was closed,
[12445] Window WA300 was
closed,
[12453] Window WA100 was
closed,
[12496] FanActuator Fan100 was
set to Stoped,
[12520] FanActuator Fan200 was
set to Stoped,
[12538] FanActuator Fan300 was
set to Stoped,
[12572] HeaterActuator
Heater100 was set to
MediumHigh,
[12587] HeaterActuator
Heater200 was set to Low,
[12607] HeaterActuator
Heater300 was set to High,
[12632] HeaterActuator
Heater700 was set to MediumLow,
[12755] HeaterActuator
Heater500 was set to Low,
[13060] Window WA500 was
opened,
[13161] Window WA500 is already
opened,
[13262] Window WA500 is already
opened,
[13324] HeaterActuator
Heater700 was set to High,
[13363] Window WA500 is already
opened,
[13465] Window WA500 is already
opened,
```

Window actuators: WA100

Temperature sensors: EXT501

Value: 18.1118192505б

Fan actuators: Fan700

Active state: Stoped

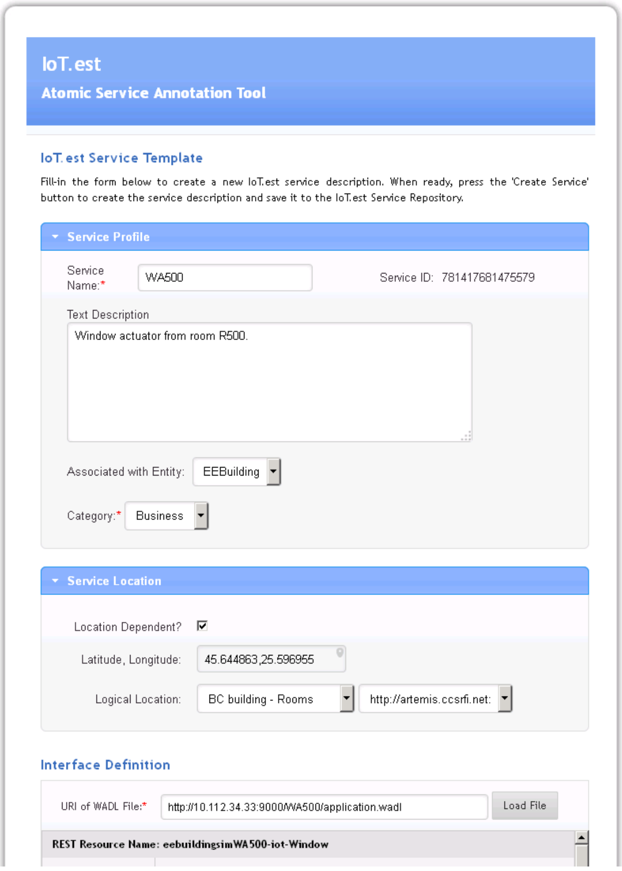
Heater actuators: Heater700

Active state: High

Current tick: 13927

Figure 16 - Front end of the EEBuildingSim simulator

The proof of concept demonstrator goes through the entire proposed lifecycle and uses all components in an integrated manner. One of the first steps is the annotation of atomic services which is performed using the Service Annotation tool (SAT) whose front end is depicted in figure 16.


IoT.est

Atomic Service Annotation Tool

IoT.est Service Template

Fill-in the form below to create a new IoT.est service description. When ready, press the 'Create Service' button to create the service description and save it to the IoT.est Service Repository.

Service Name:* WA500

Service ID: 781417681475579

Text Description: Window actuator from room R500.

Associated with Entity: EEBuilding

Category:* Business

Location Dependent? ☑

Latitude, Longitude: 45.644863,25.596955

Logical Location: BC building - Rooms | http://artemis.ccsrfi.net:

Interface Definition

URI of WADL File:* http://10.112.34.33:9000/WA500/application.wadl

REST Resource Name: eebuildingsimWA500-iot-Window

Figure 17 - Screenshot of the Service Annotation Tool (SAT)

Once the semantic description of the atomic services is done, a new composite service can be developed. This activity is centred around the Service Composition Environments (SCE) which provides: a graphical editor for business processes and their associated composite services, an atomic service search mechanism, integration with the test environment as well as with the execution environment.

Figure 18 depicts the SCE where a very simple composite service is being developed.

The same editor can be used to compose much more complex services such as the one for the WellBeing scenario which is depicted in figure 19.

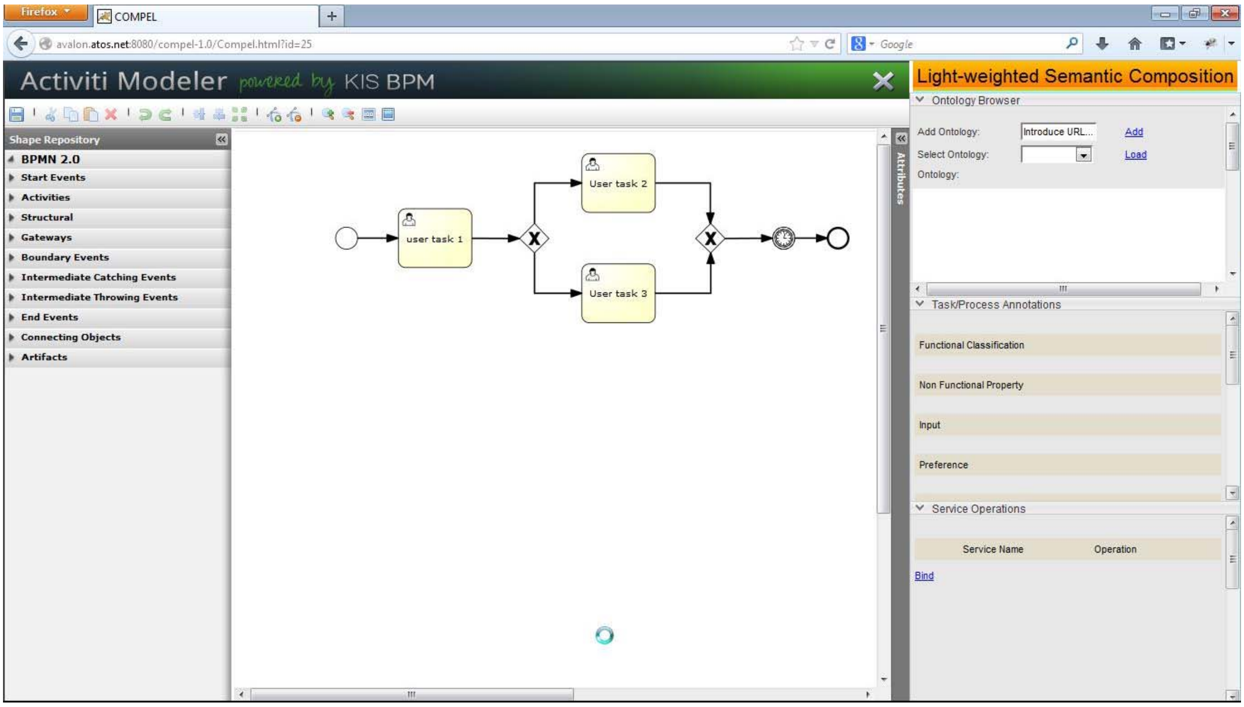


Figure 18 - Screenshot of the Service Composition Environment (SCE)

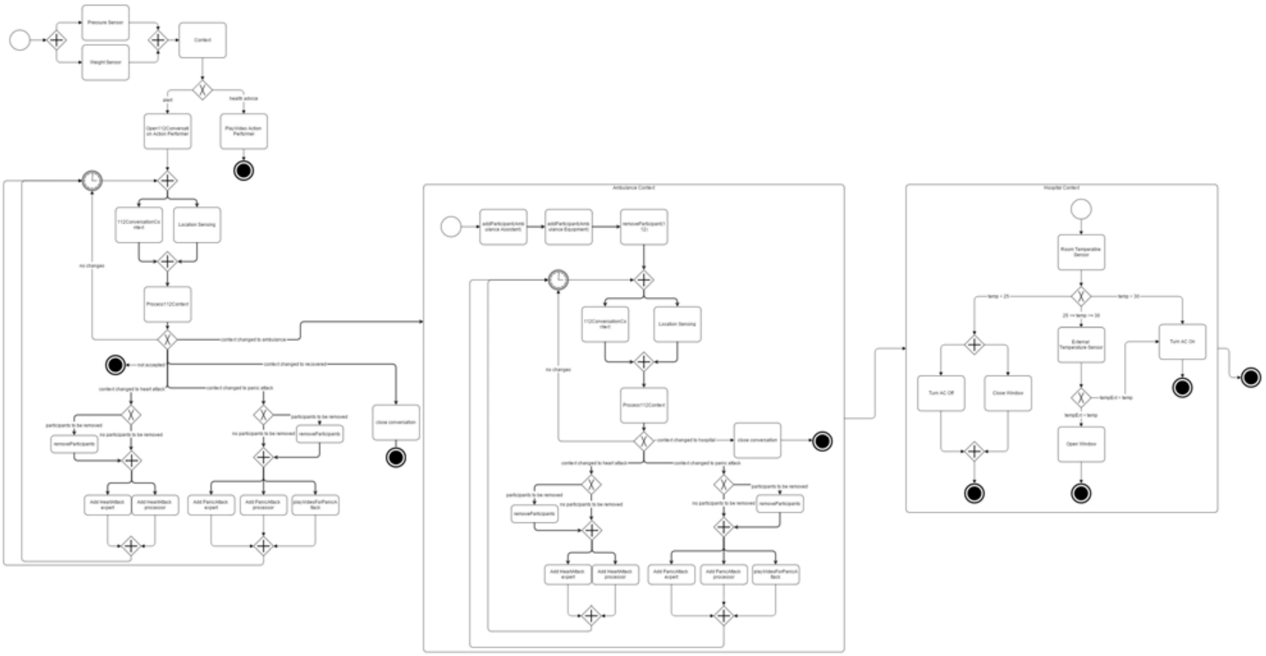
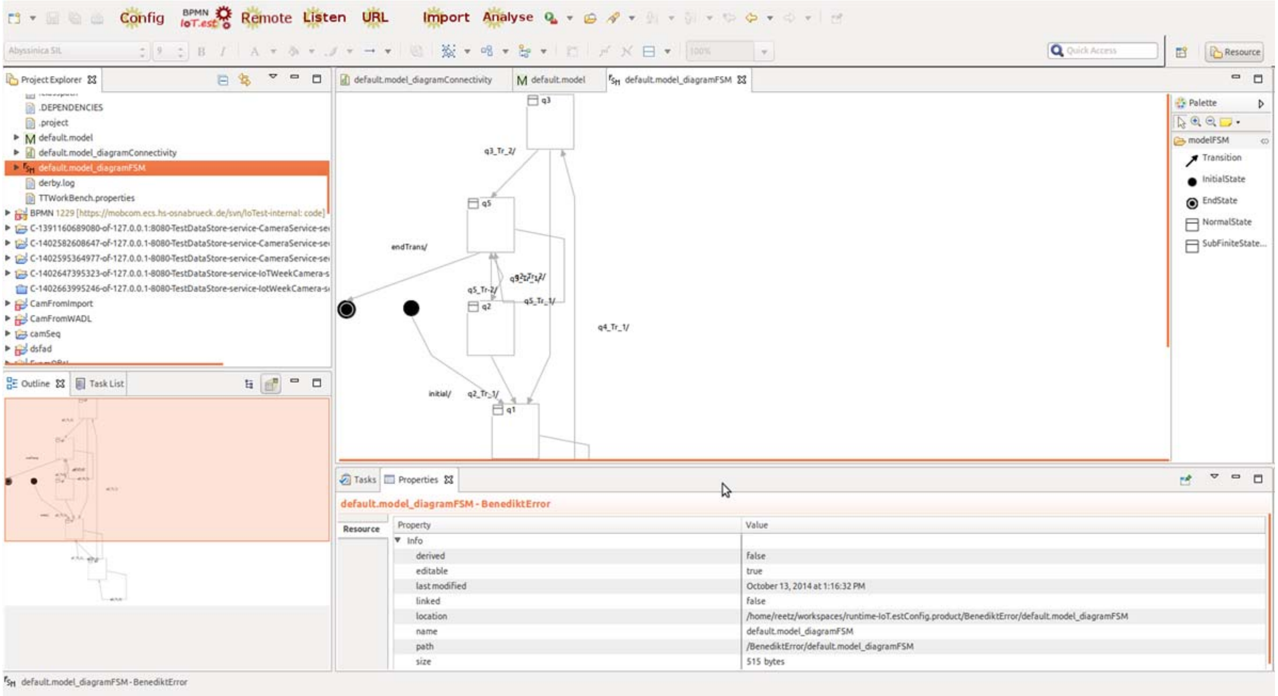


Figure 19 - Composite service of the WellBeing scenario

Test development is supported by the test component which is based on the TTWorkbench tool as seen in figure 20. Atomic and composite service testing is also integrated into the SCE which triggers the testing process once a composite service is prepared for deployment.



default.model_diagramFSM - BenediktError

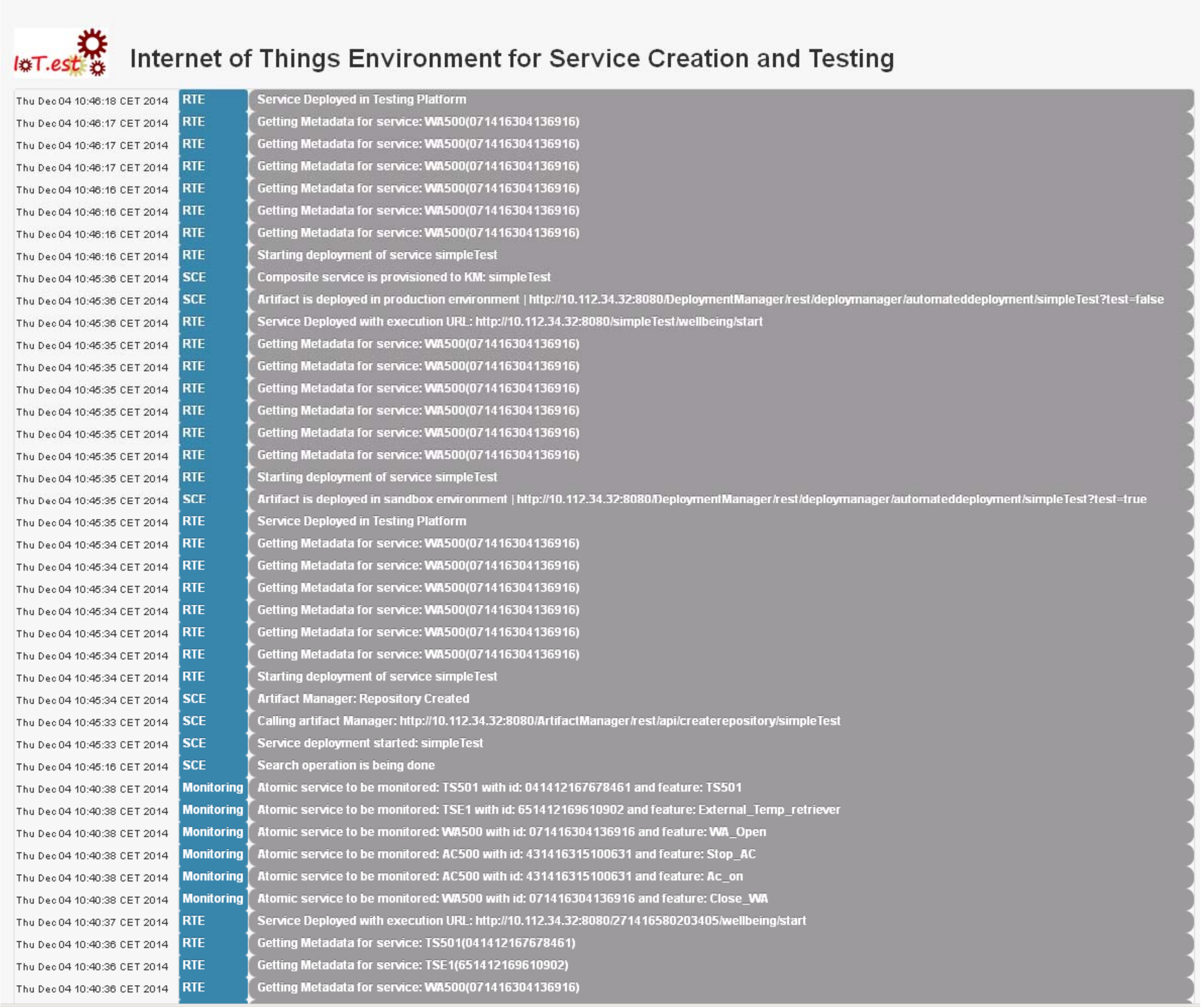
Property	Value
Info	
derived	false
editable	true
last modified	October 13, 2014 at 1:16:32 PM
linked	false
location	/home/reetz/workspaces/runtime-IoT.estConfig.product/BenediktError/default.model_diagramFSM
name	default.model_diagramFSM
path	/BenediktError/default.model_diagramFSM
size	515 bytes

default.model_diagramFSM - BenediktError

Figure 20 - GUI of the test component

The lifecycle is supported also by a number of components which do not expose any GUI but are tightly coupled as well. The Runtime Environment (RTE) is the one responsible for service deployment and for the actual execution. The Test Derivation Environment is one of the components responsible for service testing while the Monitoring component is responsible for gathering Key Performance Indicators (KPIs) from platform as well as service level. Service KPIs are involved into the automatic service replacement mechanism which is implemented by the Adaptation component.

A Logger service has also been developed to support the development and various demos. Its depiction in figure 21 shows some events related to the service monitoring and adaptation functionality.



Internet of Things Environment for Service Creation and Testing

```
Thu Dec 04 10:46:18 CET 2014  RTE         Service Deployed in Testing Platform
Thu Dec 04 10:46:17 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:46:17 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:46:17 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:46:16 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:46:16 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:46:16 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:46:16 CET 2014  RTE         Starting deployment of service simpleTest
Thu Dec 04 10:45:36 CET 2014  SCE         Composite service is provisioned to KM: simpleTest
Thu Dec 04 10:45:36 CET 2014  SCE         Artifact is deployed in production environment | http://10.112.34.32:8080/DeploymentManager/rest/deploymanager/automateddeployment/simpleTest?test=false
Thu Dec 04 10:45:36 CET 2014  RTE         Service Deployed with execution URL: http://10.112.34.32:8080/simpleTest/wellbeing/start
Thu Dec 04 10:45:35 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:35 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:35 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:35 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:35 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:35 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:35 CET 2014  RTE         Starting deployment of service simpleTest
Thu Dec 04 10:45:35 CET 2014  SCE         Artifact is deployed in sandbox environment | http://10.112.34.32:8080/DeploymentManager/rest/deploymanager/automateddeployment/simpleTest?test=true
Thu Dec 04 10:45:35 CET 2014  RTE         Service Deployed in Testing Platform
Thu Dec 04 10:45:34 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:34 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:34 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:34 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:34 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:34 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
Thu Dec 04 10:45:34 CET 2014  RTE         Starting deployment of service simpleTest
Thu Dec 04 10:45:34 CET 2014  SCE         Artifact Manager: Repository Created
Thu Dec 04 10:45:33 CET 2014  SCE         Calling artifact Manager: http://10.112.34.32:8080/ArtifactManager/rest/api/createrepository/simpleTest
Thu Dec 04 10:45:33 CET 2014  SCE         Service deployment started: simpleTest
Thu Dec 04 10:45:16 CET 2014  SCE         Search operation is being done
Thu Dec 04 10:40:38 CET 2014  Monitoring  Atomic service to be monitored: TS501 with id: 041412167678461 and feature: TS501
Thu Dec 04 10:40:38 CET 2014  Monitoring  Atomic service to be monitored: TSE1 with id: 651412169610902 and feature: External_Temp_retriever
Thu Dec 04 10:40:38 CET 2014  Monitoring  Atomic service to be monitored: WA500 with id: 071416304136916 and feature: WA_Open
Thu Dec 04 10:40:38 CET 2014  Monitoring  Atomic service to be monitored: AC500 with id: 431416315100631 and feature: Stop_AC
Thu Dec 04 10:40:38 CET 2014  Monitoring  Atomic service to be monitored: AC500 with id: 431416315100631 and feature: Ac_on
Thu Dec 04 10:40:38 CET 2014  Monitoring  Atomic service to be monitored: WA500 with id: 071416304136916 and feature: Close_WA
Thu Dec 04 10:40:37 CET 2014  RTE         Service Deployed with execution URL: http://10.112.34.32:8080/271416580203405/wellbeing/start
Thu Dec 04 10:40:36 CET 2014  RTE         Getting Metadata for service: TS501(041412167678461)
Thu Dec 04 10:40:36 CET 2014  RTE         Getting Metadata for service: TSE1(651412169610902)
Thu Dec 04 10:40:36 CET 2014  RTE         Getting Metadata for service: WA500(071416304136916)
```

Figure 21 - GUI of Logger service showing events related to service monitoring and adaptation

The components which were developed during the project have reached different levels of maturity but where completely integrated into a single and functional solution which was used for the proof of concept. The entire solution has also been evaluated along with the proposed lifecycle by persons external to the project team. The results aggregating a set of qualitative and quantitative measurements were documented in deliverable *D6.3 - Evaluation Report of Test-driven Service Creation and Provision.* Despite the lower score related to the usability of the components, the overall feedback was positive demonstrating the value of the proposed lifecycle and of the core functionality.

3 Accelerating IoT.est Impact

The IoT.est approach allows the systematic integration of IoT service development within the IoT business processes and provides a complete set of solutions to support the service lifecycle management and production line of the IoT services. This way it introduced a new breed of superior IoT enabled business processes. They are better designed, better implemented and better maintained as a result of IoT.est. The sandbox testing before the deployment phase introduces mechanisms to construct reliable IoT services using different design-time tests. Deployment test and run-time monitoring facilitates provisioning adaptable and fault tolerant services in IoT platforms. Having test-aware, adaptable and reliable business services composition and life cycle management solutions as developed in IoT.est introduces a new range of IoT enabled business processes that will transform the IoT service and application development beyond the boundaries of any specific sector .

The solutions built by IoT.est allow developers to test and explore their ideas and designs at a much earlier stage than possible before. Validation will allow them to identify both winning strategies and flawed approaches early on in the IoT service and business process developments. It will allow them to discard erroneous hypotheses and concentrate time and investment on approaches proven to work. It will allow them to test multiple innovative technologies together at an early stage, reducing later interoperability issues and taking advantage of developments in complementary fields. This will lead to quicker innovation cycles and accelerated development. Validation will lead to an improved quality of results which will make exploitation of them quicker. This will in turn provide European ICT businesses with the ability to develop more advanced IoT enabled services and products and reduce the time to market of them.

Throughout the project life time, IoT.est has played a very active role in disseminating its results and engaging in the IERC (European Research Cluster on the Internet of Things), IoT and wider ICT communities. 27 scientific papers based on IoT.est research have been published; 32 presentations and workshops have been given and 2 demonstrations performed at key conferences and special sessions.

IERC
European Research Cluster
on the Internet of Things

IoT Week 2014
16-20 June - London, UK

ETSI
World Class Standards

FUTURE
INTERNET
ASSEMBLY

3.1 Standardisation Activities

IoT.est was also involved with the following standardisation activities:

ETSI – European Telecommunications Standards Institute

ETSI is working on European telecommunication standards such as HiperLAN or Terrestrial Trunked Radio (TETRA). ETSI has also standardised the Global System for Mobile communications (GSM) cell phone system. ETSI is divided in several Technical Committees (TCs).

ETSI TC Methods for Testing and Specification (MTS)

As the origin of the standardized test specification language TTCN-3 ETSI's technical committee called Methods for Testing and Specification (MTS) is still in charge for the maintenance and extension of the standard. There is an active and ongoing change request process including a working maintenance team in place that could be utilized for possible standardization activities regarding TTCN-3.
IoT.est provided a contribution to ETSI MTS, proposing an extension of the TTCN-3 standard for delayed data evaluation to better support specific testing needs of IoT architectures. The proposal was accepted as an additional work item for the ETSI specialist task force TTCN-3 Evolution 2013 (STF460) and Testing Technologies was appointed as rapporteur for the work item. The extension passed ETSI remote consensus in March 2014, and has been published in June 2014 as part of TTCN-3 edition 4.6.1 "ETSI ES 201 873-1 V4.6.1 (2014-06) ETSI Standard Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language"
(http://www.etsi.org/deliver/etsi_es/201800_201899/20187301/04.06.01_60/es_20187301v040601p.pdf)

ETSI TC M2M / oneM2M

The standardization of M2M promotes the end of the current fragmented solutions, dependent on the specific verticals, being essential for the overall development of the M2M market. The M2M technology encompasses several areas, ranging from devices to applications, taking into account the different wireless networks. Consequently, there are several regulatory trends that intersect at certain points of the M2M domain.
It is important to highlight the work being carried by the ETSI Technical Committee M2M (TC M2M) which has defined a horizontal service architecture based on REST paradigm, which exposes a set of capabilities that enables the exchange of information between applications and devices through standardized procedures. This innovative approach promotes the existence of a common layer used for data mediation, ending with the typical vertical silos of different domains, ensuring a complete independence between applications and devices [ETSI.M2M].

Moreover, the oneM2M initiative has just announced its initial candidate release of technical specifications for Machine-to-Machine communications and the Internet-of-Things. It is focusing its normative efforts in the service layer, where a common set of service functionalities are being defined, aiming at consolidating the approaches followed by the different standards developing organisations [oneM2M].

ITU-T – International Telecommunication Union

ITU is an agency of the United Nations which was founded in 1865 to regulate information and communication technology issues. The ITU is active in areas including broadband Internet, latest generation wireless technologies, aeronautical and maritime navigation, radio astronomy, satellite based meteorology, convergence in fixed-mobile phone, security, Internet access, data, voice, TV broadcasting, and next-generation networks. The Telecommunication Standardisation Sector (ITU-T) coordinates standards for telecommunications on behalf of the ITU. The ITU-T Study Group 13 has recently started discussions on standardization of Internet of Things and Future Networks related architectures, functions and protocols.

IoT-GSI has been established in ITU in 2011 with the objective of co-ordinating IoT related standardization work of different study groups of ITU as well as of other SDOs. One of IoT.est project members from NICT has joined several meetings of IoT-GSI as a rapporteur of Q.15 of the related study group 13, which is pursuing standardization work of network architectures and protocols relevant to IoT.

OGC – Open Geospatial Consortium

Open Geospatial Consortium is a non-profit standardisation organisation that is leading the development of standards for geospatial and location based services.

IoT.est has reviewed the OGC's Sensor Web Enablement (SWE) initiative's Observations and Measurements XML Schema (O&M) [OGC O&M] while designing the reusable services of Measurement Storage Service and Measurement Retrieval Service. The O&M XML Schema can be applied for encoding both real-time and archived O&M data from a sensor. It specifies the properties of an observation including the observed property, the feature of observation and the sampling time. A similar strategy has been applied while storing the IoT device's measurement along with the recorded timestamp. This enables the Measurement Retrieval Service to allow searching for measurements within a given time interval.

W3C – World Wide Web Consortium

W3C is an international community that develops standards to support the long-term growth of the Web. W3C have built several working groups to write technical reports which are used to promote the development of standards based on community consensus. W3C interest groups are founded to bring people together and exchange ideas and innovative concepts.

The relevant W3C topics for the IoT.est project are:

SPARQL Query Language for RDF: the SPARQL language has been employed for performing semantic search within the distributed repository framework developed within the IoT.est project for storing semantic service descriptions. The semantic search process forms the second step of the discovery mechanism developed within WP3 of the project.

W3C Semantic Sensor Network Incubator Group (SSN-XG) and W3C Semantic Sensor Networks Business Community: W3C's Incubator Group on Semantic Sensor Networks[1] (SSN) defined the SSN ontology [Compton 2012] to describe sensors and sensor networks. SSN ontology defines sensor devices, properties of sensors, capabilities of sensors, systems, processes, and methods of sensing etc. The current SSN ontology is considered while designing the required information models and ontologies in WP3. Particularly, in the case where the IoT Resource concept is of type sensor, the associated concept links to the ssn:Sensor class using the Linked Data principles. The IoT.est project has been involved in the Semantic Sensor Networks Business Community group[2].

---

[1] http://www.w3.org/2005/Incubator/ssn/
[2] http://www.w3.org/community/ssn-cg/wiki/SSN_Applications

3.2 IoT.est Links to other EU Projects

IoT.est also had established close links to other EU funded research and innovation projects:

www.iot-a.eu/public

www.iot-icore.eu

www.smartsantander.eu

www.iot-cosmos.eu

www.ict-citypulse.eu

www.sociotal.eu

www.i-cargo.eu

www.fi-ware.org

www.fed4fire.eu

www.soa4all.eu

Plus: ComGeneration

The IoT.est project has established and fostered direct links with other related research projects, reusing some results from past IoT projects or contributing with its own results to other ongoing or already finished ones.

A summary of these contributions is provided in the table below:

IoT.est results	Link to inputs **from** other IoT projects	IoT projects using IoT.est results
IoT.est semantic service model	Builds upon IoT-A entity-resource-service models. Deliverable D4.3 : "Concepts and Solutions for Entity-based Discovery of IoT Resources and Managing their Dynamic Associations", 2012 (http://www.iot-a.eu/public/public-documents/documents-1)	COSMOS_WP5 Deliverable:_D5.1.1 "Decentralized and Autonomous Things Management: Design and open specification (Initial)", 2014.
Large-scale sensor service discovery	Extends IoT-A Geographic Locations approach for IoT Service Resolution, documented in Deliverable D4.3 (http://www.iot-a.eu/public/public-documents/documents-1) and D4.4: "Final Design and Implementation Report"	
WebRTC Service	WONDER project https://github.com/hypercomm/wonder	
IoT.est Test Component, Graphical Editor and EMF Model	Build upon Test Framework for communication protocol testing based on the ComGeneration project (http://www.ecs.hs-osnabrueck.de/27633.html) M. Fischer, R. Tönjes, R. Lasch: "A New Approach for Automatic Generation of Tests for Next Generation Network Communication Services", accepted for 6th IEEE International Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE 2011), Toulouse, France, 5.-9. Sep. 2011.	FP7 ICT CityPulse intends to reuse model-based service descriptions and test derivation and PBMN-based behaviour description of the IoT.est project.
Service Composition Environment (SCE)	Built upon an already existent asset from ATOS, yourBPM, the logic of the goal-oriented and knowledge-based service composition was developed within SOA4All project.	The SCE has been extended for adaptation to other domains, also outside of the IoT, and is being reused in projects such as: • i-Cargo, as a service

		orchestrator for logistics. • Fi-Ware, to be reused as a Generic Enabler. • Fed4Fire, as a service orchestrator. • CloudSocket, this project is already starting.

All of the project's dissemination activities and interactions can be found on the project website:

www.ict-iotest.eu

3.3 Use of IoT.est results

As stated before, the IoT.est project provides several results grouped by functionalities in several components. The list of final released components, their owners and their licenses can be seen in the table below:

Component	Owner	License	License URL
Knowledge Management	UniS / SIE	Apache 2.0	http://www.apache.org/licenses/LICENSE-2.0
Service Annotation Tool	UniS	Apache 2.0	http://www.apache.org/licenses/LICENSE-2.0
Service Composition Environment	ATOS	GPLv3	http://www.gnu.org/licenses/gpl.html
Test Derivation Engine	UASO	Closed academic license with partner university	N/A
Test Execution Engine	TT	Commercial	N/A
Runtime Environment	PTIN	Apache 2.0	http://www.apache.org/licenses/LICENSE-2.0
Monitoring & Adaptation	ATOS	GPLv3	http://www.gnu.org/licenses/gpl.html
	SIE	Apache 2.0	http://www.apache.org/licenses/LICENSE-2.0

Interested parties, developers, researchers, can access and download these components through the project website (http://www.ict-iotest.eu), or directly via our support site at: http://ict-iotest.eu/iotest/support this includes license and installation instructions. A quick start guide, or user manual, can be also downloaded from there.

For more information, contact the support team: iotest-support@ict-iotest.eu

3.4 The IoT.est Consortium

University of Surrey (UniS)

The Institute of Communication Systems (ICS) is a leading centre for communication research in the EE department of the Faculty of Engineering and Physical Sciences (FEPS) at The University of Surrey (UniS), UK. The prominent role that the University and the Centre maintain in the field of fixed, mobile and wireless communications is recognized from both academic and industrial point of view. The University has been recently awarded as the 8th best university in the UK and its EE department, to which the CCSR belongs, is currently ranked 2nd in the UK, according to the Guardian league table. CCSR has expertise in IoT platforms and IoT services, Knowledge Engineering and modelling of IoT resources and services, and service creation and service adaptation in mobile environments through its involvement in various EU and UK projects.

The main focus of the University of Surrey team in IoT.est was on supporting goal-oriented and knowledge-driven service creation and provisioning through Semantic Web technologies. The university has contributed to the areas of knowledge representation through interoperable description models, analysis of the semantic description of the Internet of Things-enabled services, environment and platform data to create methods for publication, distributed discovery mechanisms and run-time adaptation of the services. The goal-oriented service creation enablers facilitate dynamic binding of matching services to workflow, integration of service testing in the service creation process and a dynamic deployment configuration based on semantic platform recommendation.

Atos Spain, SA (ATOS)

Atos SE (Societas Europaea) is an international information technology services company with 2013 annual revenue of EUR 8.6 billion and 76,300 employees in 52 countries. Serving a global client base, it delivers IT services through Consulting & Systems Integration, Managed Operations, and transactional services through Worldline, the European leader and a global player in the payment services industry. With its deep technology expertise and industry knowledge, it works with clients across different business sectors: Manufacturing, Retail & Transportation; Public sector & Health; Financial Services; Telcos, Media & Utilities.

Atos is focused on business technology that powers progress and help organizations to create their firm of the future. It is the Worldwide Information Technology Partner for the Olympic and Paralympic Games and is listed on the NYSE Euronext Paris market. Atos operates under the brands Atos, Atos Consulting, Worldline and Atos Worldgrid (atos.net).

Innovation it's in our DNA is part of the company motto. Our goal is to search for innovative products, service and processes to stay one step further from our competitors. The Research and Innovation group of ATOS (www.atosreseach.eu) is the R&D node of Atos and a worldwide point of reference for the company. In 2010, the group became part of the Global Innovation, Business Development and Strategy (GIBS) Team, which is the pro-active business growth think-tank of ATOS. With its extensive experience of more than 25 years in

R&D, ARI drives the research activities on the latest technologies with results transferred to clients through commercial projects. The multi-disciplinary team is composed of experts from different knowledge areas as well as nontechnical areas. The IoT.est project is managed by the ARI division. Within ARI, the Software Engineering lab, built upon more than ten years of experience performing insightful research in service engineering and distributed systems. In the context of engineering there is an emphasis on providing state-of-the-art tools that help tomorrow's software.

Lately, to address the need imposed by our ambitious sustainability and corporate responsibility objectives, the lab has started to develop a set of assets that will drive improvement software engineering techniques and Quality of Service, taking into account its close relation to cloud computing, applications behaviour and requirements.

Alexandra Institute (AI)

The Alexandra Institute is a non-profit SME working with application-oriented ICT research. The Alexandra Institute's focus is on pervasive computing, and how to activate the business potential of customers through research-based user driven innovation. The Alexandra Institute is an Approved Technological Service Institute operating within the Danish Advanced Technology Group, a network of independent Danish research and technology organizations.

The Alexandra Institute has extensive experience with international collaboration and is involved in many international projects, collaborating with over 200 international organisations in over 20 different countries - from very small SMEs to big international companies and knowledge institutions. Some of the recent research projects include: Smart Santander (FP7) proposes a unique in the world city-scale experimental research facility in support of typical applications and services for a Smart City. The Airfield Monitor (R4SME) project aims to develop a novel system to support planning, control and assessment of airport runway and taxiway conditions, while significantly enhancing aviation safety.

National Institute of Information and Communication Technology (NICT)

NICT is the sole national research organization in this research field in Japan. The New Generation Network Research Centre focuses on both network architecture and fundamental technologies of new generation networks (NWGN), which is the future Internet vision of Japan. Under the umbrella of NWGN, NICT researches on the all emerging aspects/areas of future networks including IoT area. NICT provided the IoT.est consortium with the research experiences with regard to wireless networking, network architecture and trust/security/privacy system design.

Portugal Telecom Inovação (PTIN)

PT Inovação e Sistemas is a Portugal Telecom technological company, focused on the development of innovative products and services and in systems integration for Telco and IT markets. Its mission is to provide transformation and technological innovation to customers, allowing them to minimize Time To Market while creating advanced leading edge service offers.

PT Inovação e Sistemas pursues innovation – at the services, technologies and operations level – by developing several skills in the Telco and ICT market sectors. The company promotes the cooperation with Universities and other I&D institutes worldwide, positioning itself as a true knowledge provider, both in the market and in the industry.

With 64 years of technological experience, PT Inovação e Sistemas is an agent working for technological innovation and transformation, looking to improve people's lives and companies' processes. It has 2000 workers, most of them Telco experts and IS researchers.

Based in Aveiro, PT Inovação e Sistemas has a network of subsidiaries performing in Brazil, Morocco, Angola, Mozambique, Namibia, South Africa and Spain, through offices in Lisbon, Oporto, Salvador, São Paulo, Rio de Janeiro, Casablanca, Luanda, Maputo, Windhoek, Johannesburg and Madrid.

PTIsS joined the IoT.est project with high expectations that the project research results are of high value for the industry. From the technical point of view there are three particularly important aspects for PTInS: i) using IoT.est runtime service monitoring and adaptation mechanisms in large scale service networks; ii) using IoT.est service creation environment to promote a business driven IoT services market; iii) integrating reusable IoT services in PT service delivery domain.

SIEMENS (SIE)

SIEMENS is a global powerhouse in electric and electronic industry, a world leader in complex infrastructures solutions and an active provider of sustainable green technologies. Siemens Corporate Technology (CT) develops hardware and software systems and solutions and a broad range of services for the entire field of information and communication technologies, acting as main driver of innovation inside Siemens.

SIEMENS CT Romania, with over 250 engineers and researchers is the Romanian branch of these activities. CT has an extensive expertise and experience in all areas of software development, working for a large variety of sectors like energy, automotive and industrial manufacturing or services to mention but a few. Current research capacity and expertise of the department in Brasov (acting inside Siemens SRL, Romanian house company of Siemens) encompasses fields like Constraint Based Configurations and Schedulers, Complex Event Processing, Medical Informatics, Intelligent Networks, Sensor Networks, Data Acquisition Architectures and Future Internet Technologies, and also Application Specific Integrated Circuit (ASIC) design and verification, hardware coded security, PCB design and implementation and full capabilities to do Hardware, Software and Mechanic co-designed solutions.

The department acts at a global scale inside Siemens Corporate Technology, the core technological house of the corporation. Due to this unique position, the group contributes in the large mesh of knowledge of CT experts, with large access to the EU funded projects where Siemens CT already contributes. From the local group perspective, the expertise offered is centred around: Knowledge based systems (with a strong focus on Constraints Based Systems and the mix with Semantic technologies), Event Processing and Event Driven Architectures, Multi-Agent Systems, Big Data analytics, Cloud Computing and Networking technologies, embedded systems, digital security, Intelligent Energy Grid Management or Cities Infrastructures.

Testing Technologies (TT)

Testing Technologies IST GmbH in Berlin, Germany, is a Spin-Off of the Fraunhofer Institute for Open Communication Systems (FOKUS) and develops and markets highly innovative test development tool series, solutions and services based on the test specification and implementation language TTCN-3 that has been standardised by the

European Telecommunications Standards Institutes (ETSI). TestingTech.

Testing Tech offers under the brand TTworkbench a highly integrated test development and execution tool series for a wide range of industry domains, including the telecommunications, automotive and financial domains. TTworkbench supports the entire life cycle of TTCN-3 based tests with textual and graphical editors, a TTCN-3 to Java compiler, and a test execution management environment composed of graphical tracing, debugging and reporting facilities for centralised and distributed test components. TTworkbench supports different system modelling languages such as UTP, IDL, ASN.1, WSDL and XML. Besides the generic tool series, Testing Tech provides also domain specific solutions in the area of IMS, such as TTsuite-SIP, a comprehensive test suite for voice over IP technologies. In addition, tailored development services can be provided to the customer, as well as training for testing methods and tools. Furthermore, Testing Tech is an active contributor to standardization processes in international consortia, in particular in ETSI.

Testing Tech has been part of the successful ITEA TTmedal and ITEA2 D-MINT projects. The tool series could be evaluated and extended in particular throughout the industrial case studies of the projects.

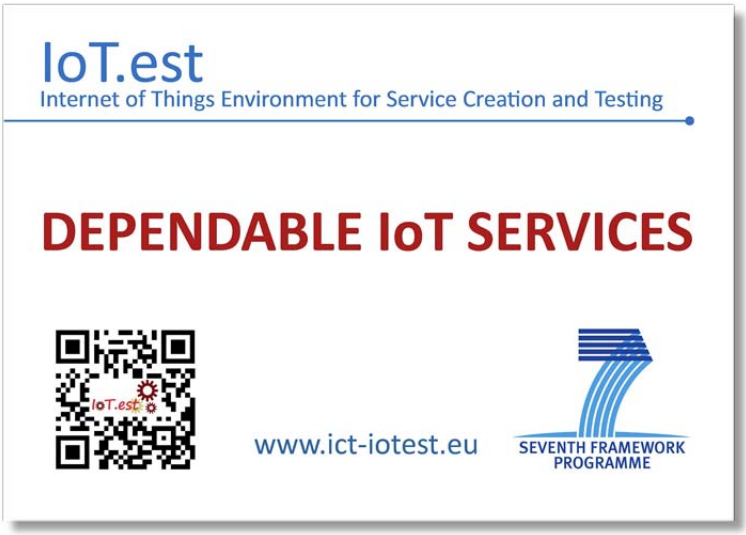
The main focus of the Testing Technologies team in IoT.est is on supporting IoT related requirements in their TTCN-3 tool series. This comprises contribution to definition of test architecture, definition of test components and test interfaces and to the development of the test environment and test execution framework.

University of Applied Sciences Osnabrück (UASO)

Fachhochschule Osnabrück
University of Applied Sciences

University of Applied Sciences Osnabrück (UASO) is a public foundation which is authorised to administer itself. The faculty of engineering and computer science is headed by 73 professors. Close co-operation with business and industry and scientific know-how combined with practical experience in one of more than 80 laboratories guarantee innovative research results relevant for the praxis in business and industry. It has participated in the European SOKRATES/ERASMUS and LE- ONARDO DA VINCI programmes and several EU research projects, such as INCO, NNE-JOULE and EQUAL, C-MOBILE, C-CAST and MORYNE. The mobile communication group has been active on various aspects of analysis and design of wireless systems. The major focus has been on performance analysis of network protocols, simulation at link and system level and service platforms. Its in- depth knowledge is reflected by many scientific publications. Since ten years the group organizes the German national Mobile Communication Workshop (Mobilkomtagung) once a year. The laboratory is equipped with several simulation tools, network analysers, and a test network for development and demonstration of Service Oriented Architectures.

In the IoT.est project, UASO provides its research experiences, in particular with regard to mobile communication systems and designing test systems for communication services. This leads to the knowledge driven model based testing approach implemented in the project. Moreover, IoT.est benefits from UASO’s know how about service creation environments and service oriented architectures.



IoT.est

Internet of Things Environment for Service Creation and Testing

DEPENDABLE IoT SERVICES

www.ict-iotest.eu

SEVENTH FRAMEWORK PROGRAMME

4 Use and Dissemination of Foreground

4.1 Publications

List of scientific (peer reviewed) publications

NO.	Title	Author(s)	Publisher	Place of publication	Year of publication	Permanent identifiers	Is/Will open access provided to this publication?
1	*"A Comprehensive Ontology for Knowledge Representation in the Internet of Things"*	*Wei Wang, Suparna De, Ralf Tönjes, Eike Reetz, Klaus Moessner,*	*IEEE*	*KAMIoT 2012 in conjunction with IEEE IUCC-2012, Liverpool, UK*	*June, 2012*	http://epubs.surrey.ac.uk/716727/	Yes
2	*"A Test-driven Approach for Life Cycle Management of Internet of Things enabled Services "*	*Ralf Tönjes, Eike Reetz, Klaus Moessner, Payam Barnaghi*	*Future Network & Mobile Summit*	*Future Network & Mobile Summit 2012, Berlin, Germany*	*July, 2012*	http://epubs.surrey.ac.uk/736928/	Yes
3	*“PDAF: Proactive Distributed Authentication Framework for Regional Network”,*	*Ruidong Li, Masaaki Ohnishi, Yasunori Owada, and Hiroaki Harai*	*IEEE*	*The 7th Asia Joint Conference on Information Security (AsiaJCIS 2012), Tokyo, Japan*	*August, 2012*	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6298133	No

4	*“A Proactive Scheme for Securing ID/Locator Split Architecture”,*	*Ruidong Li, Ved P. Kafle, and Hiroaki Harai*	*IEEE*	*IEEE International Conference on Network Protocols 2012 (ICNP 2012) workshop on Secure Network Protocol*	*October, 2012*	http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6459950	No
5	*"Semantics for the Internet of Things: early progress and back to the future"*	*Payam Barnaghi, Wei Wang, Cory Henson, Kerry Taylor*	*International Journal on Semantic Web and Information Systems*	*International Journal on Semantic Web and Information Systems (special issue on sensor networks, Internet of Things and smart devices)*	#	http://personal.ee.surrey.ac.uk/Personal/P.Barnaghi/doc/IJSWIS_SemIoT_CR_2.pdf	Yes
6	*"Semantic Sensor Service Networks"*	*Wei Wang, Payam Barnaghi, Gilbert Cassar, Frieder Ganz, Pirabakaran Navaratnam*	*IEEE*	*Proceedings of the IEEE Sensors 2012 Conference, Taipei, Taiwan*	*October, 2012*	http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6411490	No
7	*"Computing Perception from Sensor Data"*	*Payam Barnaghi, Frieder Ganz, Cory Henson, Amit Sheth*	*IEEE*	*Proceedings of the IEEE Sensors 2012 Conference, Taipei, Taiwan*	*October, 2012*	http://epubs.surrey.ac.uk/721671/	Yes

8	*An Architecture for the Future Business of Things*	*Filipe Cabral Pinto, Paulo Chainho, Nuno Pássaro, Fernando Santiago, Daniel Corujo, Diogo Gomes*	*3rd International Conference on the Internet of Things (IoT2012)*	*3rd International Conference on the Internet of Things (IoT2012), Wuxi, China,*	*October, 2012*	-	No
9	*"A Resource Mobility Scheme for Service-Continuity in the Internet of Things"*	*Frieder Ganz, Ruidong Li, Payam Barnaghi, Hiroaki Harai*	*IEEE*	*Proceedings of the IEEE International Conference on Internet of Things (iThings 2012), Besançon, France*	*November, 2012*	http://epubs.surrey.ac.uk/771422/	Yes
10	*"A Hybrid Semantic Matchmaker for IoT Services"*	*Gilbert Cassar, Payam Barnaghi, Wei Wang, Klaus Moessner*	*IEEE*	*Proceedings of the IEEE International Conference on Internet of Things (iThings 2012), Besançon, France*	*November, 2012.*	http://epubs.surrey.ac.uk/771425/	Yes
11	*"Test Driven Life Cycle Management for Internet of Things based Services: a Semantic Approach"*	*Eike S. Reetz, Daniel Kümper, Ralf Tönjes, Anders Lehmann*	*VALID 2012*	*VALID 2012, The Fourth International Conference on Advances in System Testing and Validation Lifecycle, Lisbon, Portugal,*	*November, 2012.*	http://www.thinkmind.org/download.php?articleid=valid_2012_1_40_40101	Yes

12	*"Knowledge Representation in the Internet of Things: Semantic Modelling and its Applications"*	*Wei Wang, Suparna De, Gilbert Cassar, Klaus Moessner,*	*KoREMA - Croatian Society for Communication, Computing, Electronics, Measurement and Control*	*Automatika -Journal for Control, Measurement, Electronics, Computing and Communication, Special issue on Knowledge Acquisition and Management in the Internet of Things*	*2012*	http://epubs.surrey.ac.uk/794745/	Yes
13	*"A Distributed Hybrid Matchmaker for IoT Services"*	*Gilbert Cassar, Payam Barnaghi, Klaus Moessner*	*The International Transactions on Systems Science and Applications*	*The International Transactions on Systems Science and Applications, Special Issue on Future Internet of Things*	*2012*	http://epubs.surrey.ac.uk/771424/	Yes
14	*"Test-Enabled Architecture for IoT Service Creation and Provisioning"*	*Suparna De, Francois Carrez, Eike Reetz, Ralf Toenjes, Wei Wang*	*Springer Berlin Heidelberg*	*Book chapter in The Future Internet: Future Internet Assembly 2013: Validated Results and New Horizons*	*2012*	http://link.springer.com/chapter/10.1007%2F978-3-642-38082-2_20	Yes
15	*"How to Test IoT Services before Deploying them into Real World,"*	*Eike Reetz, Daniel Kuemper, Klaus Moessner, Ralf Tönjes*	*European Wireless Conference*	*Proceedings of the 19th European Wireless Conference (EW2013), Guildford, UK*	*April, 2013*	https://www.vde-verlag.de/proceedings-de/563498066.html	No

16	*The Business of Things Architecture*	*Filipe Cabral Pinto, Paulo Chainho, Nuno Pássaro, Fernando Santiago, Daniel Corujo, Diogo Gomes*	*Wiley Online Library*	*Transactions on Emerging Telecommunications Technologies, Vol. 44, No 4, pp. 441–452,*	*May, 2013*	http://onlinelibrary.wiley.com/doi/10.1002/ett.2654/abstract	No
17	*"Open Services for IoT Cloud Applications in the Future Internet"*	*Martin Serrano, Hoan Nguyen M. Quoc, Manfred Hauswirth, Wei Wang, Payam Barnaghi, Philippe Cousin*	*IEEE*	*Proceedings of the Second IEEE WoWMoM Workshop on the Internet of Things: Smart Objects and Services, IoT-SoS 2013*	*June, 2013*	http://epubs.surrey.ac.uk/797901/	Yes
18	*"Test Derivation for Semantically Described IoT Services"*	*Daniel Kuemper, Eike Reetz, Ralf Tönjes*	*IEEE*	*Proceedings of the 22nd Future Network and Mobile Summit 2013, Lisbon, Portugal*	*June, 2013*	http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6633588	No
19	*"Test-Enhanced Life Cycle for Composed IoT-Based Services"*	*Daniel Kuemper, Eike Reetz, Daniel Hölker, and Ralf Tönjes*	*Springer Berlin Heidelberg*	*Advances in Communication Networking: 19th EUNICE/IFIP EG WG 6.6 International Workshop, Chemnitz, Germany*	*August, 2013*	http://link.springer.com/chapter/10.1007%2F978-3-642-40552-5_32	No

20	*“Design and Implementation of A Proactive Distributed Authentication Framework (PDAF)”*	*Ruidong Li, Kazuyuki Morioka, Yasunori Owada, Masaaki Ohishi, and Hiroaki Harai*	*IEEE*	*IEEE International Conference on Network Protocols 2013 (ICNP 2013) workshop on Secure Network Protocol*	*October, 2013*	http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6733674	No
21	*"Information Abstraction for Heterogeneous Real World Internet Data"*	*Frieder Ganz, Payam Barnaghi, Francois Carrez,*	*IEEE*	*IEEE Sensors Journal, Volume:13 Issue:10*	*October, 2013*	http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6548071	No
22	*"Investigation of Opportunities for Test Case Selection Optimisation Based on Similarity Computation and Search-Based Minimisation Algorithms"*	*Eike Steffen Reetz, Daniel Kuemper, Klaus Moessner, Ralf Tönjes:*	*VALID 2014*	*The 6th International Conference on Advances in System Testing and Validation Lifecycle VALID 2014, Nice, France*	*October, 2014*	http://www.thinkmind.org/index.php?view=article&articleid=valid_2014_1_40_40050	Yes
23	*"From Semantic IoT-Service Descriptions to Executable Test Cases - Information Flow of an Implemented Test Framework"*	*Daniel Kuemper, Eike Steffen Reetz, Elke Pulvermueller and Ralf Toenjes*	*VALID 2014*	*The 6th International Conference on Advances in System Testing and Validation Lifecycle VALID 2014, Nice, France*	*October, 2014*	http://www.thinkmind.org/index.php?view=article&articleid=valid_2014_2_30_40055	Yes

24	*“An Experimental Study on Geospatial Indexing for Sensor Service Discovery”*	*Wei Wang, Suparna De, Gilbert Cassar, Klaus Moessner*	*Elsevier*	*Elsevier Expert Systems With Applications Journal*	*2014*	http://www.sciencedirect.com/science/article/pii/S095741741400757X	Yes
25	*"Enabling Query of Time-series Data from Mobile Sensing Sources”*	*Yuchao Zhou, Suparna De, Wei Wang and Klaus Moessner*	*IEEE*	*13th IEEE International Conference on Ubiquitous Computing and Communications (IUCC 2014)*	*2014*	Not available online yet	Yes
26	*Internet of Things Environment for Service Creation and Testing*	*Lara López*	*NESSIE*	*NESSIE Platform*	*2014*	Not available online yet	Yes
27	*IoT.est – Entorno para la composición y testeo de servicios IoT*	*Lara López*	*PLANETIC*	*PLANETIC Platform*	*2014*	Not available online yet	Yes

4.2 Other Dissemination Activities

List of dissemination activities					
NO.	**Type of activity**	**Main leader**	**Title**	**Date / Period**	**Place**
1	Presentation	UniS	"Mechanisms for Real World Services"	Oct-11	Internet of Things Workshop during the Future Internet Week, Poznan, Poland
2	Presentation	UniS	"IoTest project: Semantic interoperability"	Oct-11	Internet of Things Workshop during the Future Internet Week, Poznan, Poland
3	Presentation	UniS	"Achieving Semantic Interoperability in the Internet of Things"	Mar-12	PROBE-IT Workshop, Paris, France
4	Presentation	UniS	"Interoperability issues and challenges for IoT Services and Resources"	Mar-12	PROBE-IT Workshop, Paris, France
5	Presentation	ATOS	"Evolving the way we create and test services for the Internet of Things"	May-12	RCIS 2012 - Sixth International Conference on Research Challenges in Information Science, Valencia, Spain
6	Presentation	AI	"Naming, Search and Discovery in IoT: Issues and proposed solutions in the FP7 EU IoT.est Project"	May-12	IERC AC2 meeting at the FIA, Aalborg Denmark
7	Presentation	UniS	"IoT.est Project ID Card"	May-12	IERC European Research Cluster on the Internet of Things
8	Presentation	UniS	"Environment for Service Creation and Testing in the Internet of Things"	Jun-12	IERC AC14 Cognitive Technologies in IoT during the IoT-week 2012, Venice, Italy

9	Presentation	UniS	"Semantic Interoperability Issues and Approaches in the IoT.est Project"	Jun-12	IERC AC14 Cognitive Technologies in IoT during the IoT-week 2012, Venice, Italy
10	Presentation	UniS	"Architectural issues in the IoT.est Project"	Jun-12	IERC AC14 Cognitive Technologies in IoT during the IoT-week 2012, Venice, Italy
11	Presentation	SIE	"Internet of Things Environment for Service Creation and Testing (IoT.est)"	Jun-12	CompArch 2012, Bertinoro, Italy
12	Presentation	TT	"NWI proposal Data Fuzzing with TTCN-3 as a TTCN-3 Extension"	Jan-13	ETSI MTS#58, Sophia Antipolis, France
13	Presentation	UniS	"IoT.est Architecture and Mapping to IoT-A ARM"	Feb-13	IERC and Activity Chains Meetings – AC1: Architecture approaches and models, Delft, The Netherlands
14	Presentation	UniS	"Semantic IoT Semantic Inter-Operability Practices-Part 1 & 2"	Apr-13	IERC AC4 IoT Semantic Interoperability workshop, Guildford, UK
15	Presentation	TT	Testing the Internet of Things	Oct-13	1st ETSI User Conference on Advanced Automated Testing, Paris, France
16	Presentation	AI	The IoT.est Project: Description and Objectives	Jun-14	IERC Cluster Meeting, IoT Week 2014, London, UK
17	Presentation	PTIN	Evolution of Communication Infrastructures towards IoT enabled Web-based Architecture	Nov-14	5th FOKUS FUSECO Forum, Berlin, Germany
18	Presentation	PTIN	Going beyond IMS. IoT enabled Emergency Conversation Use Case.	Dec-14	WebRTC Conference Expo 2014, Paris, France

19	Poster	UniS	The IoT.est Project: Description and Objectives	Oct-11	Centre for Communication Systems Research, University of Surrey
20	Poster	UASO	Internet of Things Environment for Service Creation and Testing	May-12	17. ITG Fachtagung Mobilkommunikation, Osnabrück, Germany
21	Poster	TT	Testing the Internet of Things	Jun-12	ETSI TTCN-3 User Conference and Model Based Testing Workshop, Bangalore, India
22	Poster	All	The IoT.est Project: Demonstration	Jul-13	Future Network and MobileSummit 2013, Lisbon, Portugal
23	Poster	UASO	Test-Enhanced Life Cycle for Composed IoT-Based Services	Aug-13	19th EUNICE Workshop on Advances in Communication Networking
24	Poster	TT	Testing the Internet of Things	Oct-13	1st ETSI User Conference on Advanced Automated Testing, Paris, France
25	Poster	All	The IoT.est Project: Demonstration	Jun-14	IoT-week 2014, London, UK
26	Poster	UASO	Test Framework for IoT-Based Services: A Knowledge Driven Approach	Jun-14	23rd European Conference on Networks and Communications (EuCNC) 2014, Bologna, Italy
27	Flyer	UniS	The IoT.est Project Leaflet (General)	Oct-11	Everywhere
28	Flyer	UniS	The IoT.est Project Leaflet	Jun-12	IERC, European Research Cluster on the Internet of Things

29	Flyer	CalifThe	The IoT.est Project Booklet and Business Card	Jun-14	IoT-week 2014, London, UK
30	Workshop	ATOS	“Be Fit, Fast and Alert when testing, maintaining or solving bugs of your Future Internet applications”	May-12	FITTEST & FastFIX & ALERT & IoT.est joint workshop at IEEE RICS, Valencia, Spain
31	Workshop	UniS	IERC AC4 Semantic Interoperability Workshop	Jun-12	IoT Week, Venice, Italy
32	Workshop	UniS	The 1st International Workshop on Self-aware Internet of Things	Sep-12	Joint IoT.est, OUTSMART, iCore, and BUTLER workshop at ICAC 2012, San Jose, California
33	Workshop	UniS	Tutorial on Knowledge Engineering	Oct-12	EKAW 2012, Galway, Ireland
34	Workshop	UniS	IERC AC4 Workshop IoT Semantics Challenges and Interoperability	Oct-12	Joint IoT.est, Probe-IT and OPENIoT workshop within ETSI TC M2M demos and workshop, Mandelieu, France
35	Workshop	UniS	IERC AC4 IoT Semantics for Standardisers	Oct-12	Joint IoT.est, Probe-IT and OPENIoT workshop within ETSI TC M2M demos and workshop, Mandelieu, France
36	Workshop	UniS	Tutorial on Ontology Engineering	Oct-12	EKAW 2012, Galway, Ireland
37	Workshop	UniS	"Can Your IoT World Talk and Understand Other IoT Worlds?"	Apr-13	IERC AC4 IoT Semantic Interoperability workshop, Guildford, UK
38	Workshop	UniS	IoT Semantic Interoperability	Apr-13	IERC AC4 IoT Semantic Interoperability workshop, Guildford, UK

39	Workshop	PTIN	The 2nd International Workshop on Self-aware Internet of Things	Jun-13	Joint IoT.est, OUTSMART, iCore, and BUTLER workshop at ICAC 2013, San Jose, California
40	Workshop	ATOS	Tutorial: “Data Processing and Semantics for Advanced Internet of Things (IoT) Applications: modeling, annotation, integration, and perception”	Jun-13	WIMS-13: International Conference on Web Intelligence, Mining and Semantics, Madrid, Spain
41	Workshop	UniS	Practices on Semantics and Semantic model Interoperability	Sep-13	EUROAFRICA-ICT REGIONAL WORKSHOP, Nairobi, Kenya
42	Workshop	SIE	“Using semantic technologies to support service annotation” / “Service Monitoring and Adaptation in the IoT environment”	Feb-14	Offices of Siemens, Brasov, Romania
43	Exhibition	All	IoT.est Demonstration	Jul-13	ICT Future Network & Mobile Summit conference in Lisbon, Portugal
44	Exhibition	All	IoT.est Demonstration	Jun-14	IoT-week 2014, London, UK