

EURIDICE – An enabler for intelligent cargo for the logistics sector

Jens SCHUMACHER

**Process and Product Engineering, University of Applied Sciences
Dornbirn, 6850, Austria**

and

Manfred GSCHWEIDL

**Process and Product Engineering, University of Applied Sciences
Dornbirn, 6850, Austria**

and

Mathias RIEDER

**Process and Product Engineering, University of Applied Sciences
Dornbirn, 6850, Austria**

ABSTRACT

EURIDICE is an EU funded project which deals with the development and implementation of new concepts in the area of intelligent Cargo. The goal of EURIDICE is to develop a free and open standards based platform for intelligent goods, by using distributed mobile systems. This paper will provide an overview about the architecture and key technologies.

Keywords: Distributed intelligence, Intelligent goods, Logistics, Mobile Agents, SOA, JADE, FIPA, Collaboration platform and communication infrastructure

INTRODUCTION

Transport and logistics act in a worldwide distributed business world as an aorta of the economic system. The Logistics area has seen a huge growth in the last few years. This growth is on the one hand a result of the globalization which has led to international supply chains requiring sophisticated logistics concepts. The exponential growth of e-commerce has additionally boosted the need for logistics concept. While the overall consignment number has increased the consignment size has decreased, leading to more and smaller consignments that need to be transported to different locations. The smaller consignments pose a huge problem to the logistics services providers and their goal to keep the bundling of the consignments as long as possible in order to enable the best usage of the transport vehicles with as much consignments as possible. This goal hasn't been reached so far and thus e.g. the utilization of trucks is in some countries lower than 70% within the EU. [1] This rather weak utilization of the transport vehicles does not only pose an economic problem to the logistics services providers as well as forwarders, but also poses a substantial problem to our environment. The emissions caused by transport are still very high, as for example within the EU the transport sector accounted for 23% of total CO₂ emissions, with road transport generating 71% of total transport emissions in 2006, and are expected to increase until 2030. [1] The ultimate

goal therefore has to be to increase the utilization of the transport vehicles..

A high quality standard in this area is the basic prerequisite for the success. The investments in mobile infrastructures related to information and communication (e.g. UMTS, Galileo) allow a re-orientation of the logistics control systems to guarantee a comprehensive quality assurance of the logistics processes. Starting point for the project EURIDICE [2] is the single logistics object which should be pursued and controlled individually. By using mobile systems and communication resources logistics objects are able to reach a very high degree in self-control, which allows them to immediately react to unexpected events independently of central control structures..

The basic idea of EURIDICE is the implementation of a federative platform for information services related to intelligent cargo in the centre are the single goods and their interactions with the most different IT systems and users.

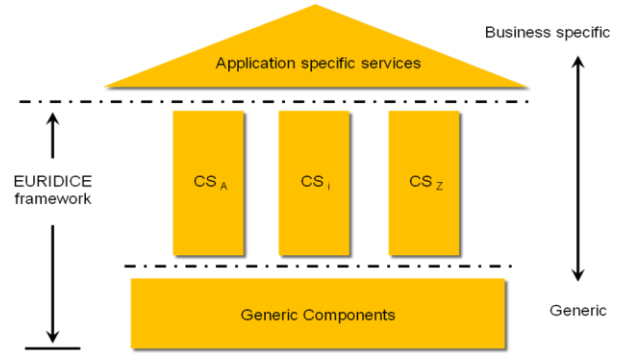
By utilization of new information and communication infrastructures and usage of the provided services, it is possible to define a uniform information triple item (time, place, status) for all transported goods in Europe. Currently this data is only used for „tracking and tracing“. EURIDICE is going to develop new value added services on this data which should allow an individual self-control of the transported goods. For an effective implementation the data acquisition and decision-making should result to the greatest extent possible on mobile devices, installed at the cargo (e.g. a box) itself, being able to react on one hand without delays to logistics events and on the other hand to reduce communication overheads.

VISION OF INTELLIGENT CARGO

The fundamental question that is related to this issue is whether it is worthwhile to retain to structured business processes or whether it would be better to define a new self-controlled business process regarding future logistics operations. The European project EURIDICE is investigating the later approach.

The project is based on the assumption that in the future the usage of passive and active RFID chips will increase and as such the availability of intelligent mobile devices will likewise increase. This leads to a situation where more and more local intelligence is available which allows the local processing of information and thus leads to local decisions on the basis of this information. EURIDICE tributed to the fact that computing capabilities are increasingly decentralized, which leads to a situation where distributed computing capabilities are used on the spot to make local decisions within the local environment and current context, instead of taking the typical hierarchical client server approach.

The core design of the platform is based on available standards like SOA (Service Orientated Architecture) and Multi-Agent-Systems (MAS). This core is enhanced with implemented services derived from extensive requirements analysis from different stakeholders in logistics chains, e.g. logistics services providers, infrastructure providers, ports and authorities and production companies (Figure 1).



CS = Common EURIDICE specific service

Figure 2: General levels and scopes of EURIDICE services

The common EURIDICE specific services (CS) allow the composition of applications specific services for the definition and orchestration of business processes on top of them. These application specific services or business processes can also be used by other application specific services and processes. On the lower levels of the framework, the generic components

ARCHITECTURE

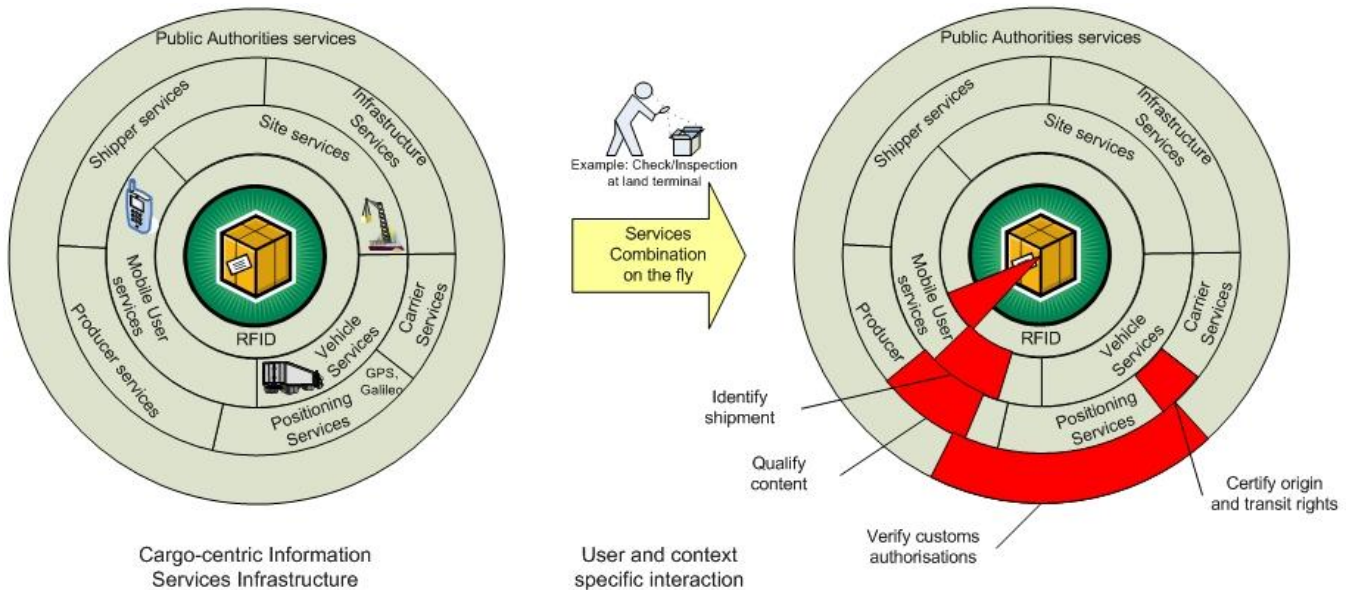


Figure 1: EURIDICE vision of intelligent cargo and value added services for different stakeholders

The project builds upon an ICT platform providing value added services on top of a SOA infrastructure. Through combination and orchestration of the available services holistic business processes can be implemented on top of the platform. By utilization of the offered services and implementation of the concepts for “intelligent cargo” the platform can react autonomously on defined events.

The framework and platform consist out of services at different levels and scopes. Roughly these services can be categorized as shown in Figure 2.

level, EURIDICE provides generic services for internal use within the framework..

The concept of a distributed and intelligent platform intends that every commodity has its own intelligence. Basically for the efficient tracking and tracing of the goods, but furthermore to receive and evaluate other relevant information regarding the transported goods on a real-time basis.

Around these single intelligent goods an open platform is created which integrates the preserved information and offered services with existing (legacy) systems. Thus the interoperability with other software products is thereby guaranteed on the basis of established standards and technologies (Figure 3).

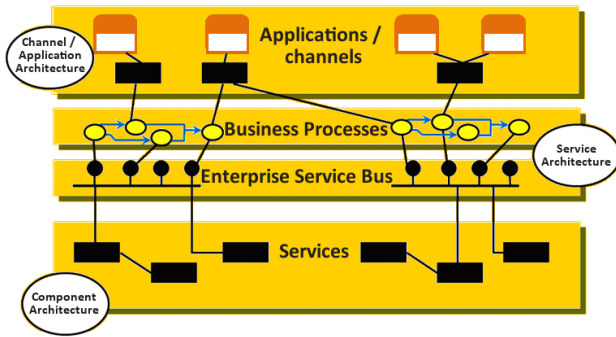


Figure 3: EURIDICE services orchestration and business process integration

The communication between the single services in the SOA is handled by an “Enterprise Service Bus” (ESB). On top of the ESB a Business Process Engine is used for the orchestration and execution of these services and the business processes needed for the different business cases.

One key issue for the easy adoption of the system is the interoperability and adaptability not only for the implemented pilot applications, but also for the future therefore open standards are a must. The realized web services are based on well known and established standards (e.g. SOAP, WSDL, UDDI, etc.). For the implementation and modeling of business processes on top of them BPEL (Business Process Execution Language) and BPMN (Business Process Modeling Notation) are used.

To achieve a close cooperation of backend systems and the transported intelligent goods, the architecture can be broken up in a centralized and a distributed part.

EURIDICE framework – centralized part

The centralized part of the EURIDICE framework consists of a set of services intended to offer central management- and infrastructure-components and to represent a common entry point for end-users. From the customers’ point of view the EURIDICE framework appears as a coherent system, where the more lower-level and distributed part is not visible to components outside the system as shown in Figure 4.

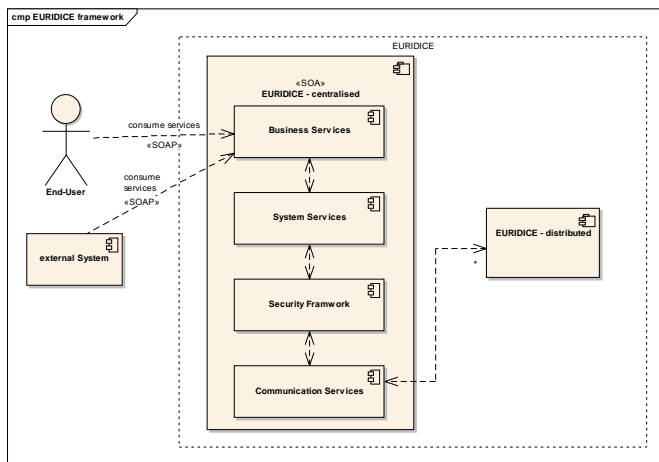


Figure 4: High level architecture of the EURIDICE framework – Customers’ point of view (UML Component Diagram)

The central part of the system, allows the integration of user defined business processes. As mentioned in the introduction of this paper, the EURIDICE system lifts single cargo items into a top-position within the business processes. Enriching the capabilities and the responsibilities of cargo items not only on a conceptual level, can be reached by equipping them with computation capabilities and integrating those parts into the business processes, which means not to just use them as a source of information, but additionally promoting them to the central part of the system. Therefore cargo-related processes are directly driven by the concerned cargo items.

EURIDICE framework – distributed

Within the EURIDICE system single cargo items are not only virtually present as a single data-set within an ERP system. Instead, transported goods become pro-active parts of the software. They control their own lifecycle and their own processes. The solution of our choice to enable cargo items to actively take part on the business-processes is to implement a Multi-Agent System [3][4] where cargo items own their virtual representation, so called agents, on intelligent devices, as described later on.

In the next section an overview about the main architecture components of the platform will be provided.

EURIDICE ARCHITECTURE COMPONENTS

In Figure 5 the main components of the EURIDICE architecture and their relationships are shown from a functional viewpoint. A more detailed description of them, their features and their role are provided in the following paragraphs. [5]

ecNodes

The abstract concept of an ecNode (EURIDICE Compliant Node - pronunciation "easy node") is used for the reduction of complexity in representation for all business objects (e.g. cargo, means of transportation, services and even human operators). An ecNode has a unique identifier and additionally has a reference to an information provider, providing additional information when authorized. Therefore global identification and discovery services, as described in the concepts of SOA, are included in the architecture for information retrieval based on an ecNode identifier.

Intelligent Cargo Network

The Intelligent Cargo Network is formed by identified cargo objects. It consists of a centralized and distributed part and provides the glue between them. It offers and defines the concept of intelligent cargo as the combination of cargo, a cargo centric approach to deal with related information and the use of associated Cargo Agents to assist in the exchange of cargo related information. The Assisting Cargo Agent (ACA) is a special kind of information provider within the Intelligent Cargo Network. It is introduced and used as a single point of access for the exchange of information about cargo, and acts like a virtual representation for an intelligent cargo item that supports information access. The references to cargo information are collected in an electronic cargo document, maintained by the Assisting Cargo Agent. The Assisting Cargo Agent represents only a purely virtually representation of the cargo, while the Operational Cargo Agent (OCA) assists the

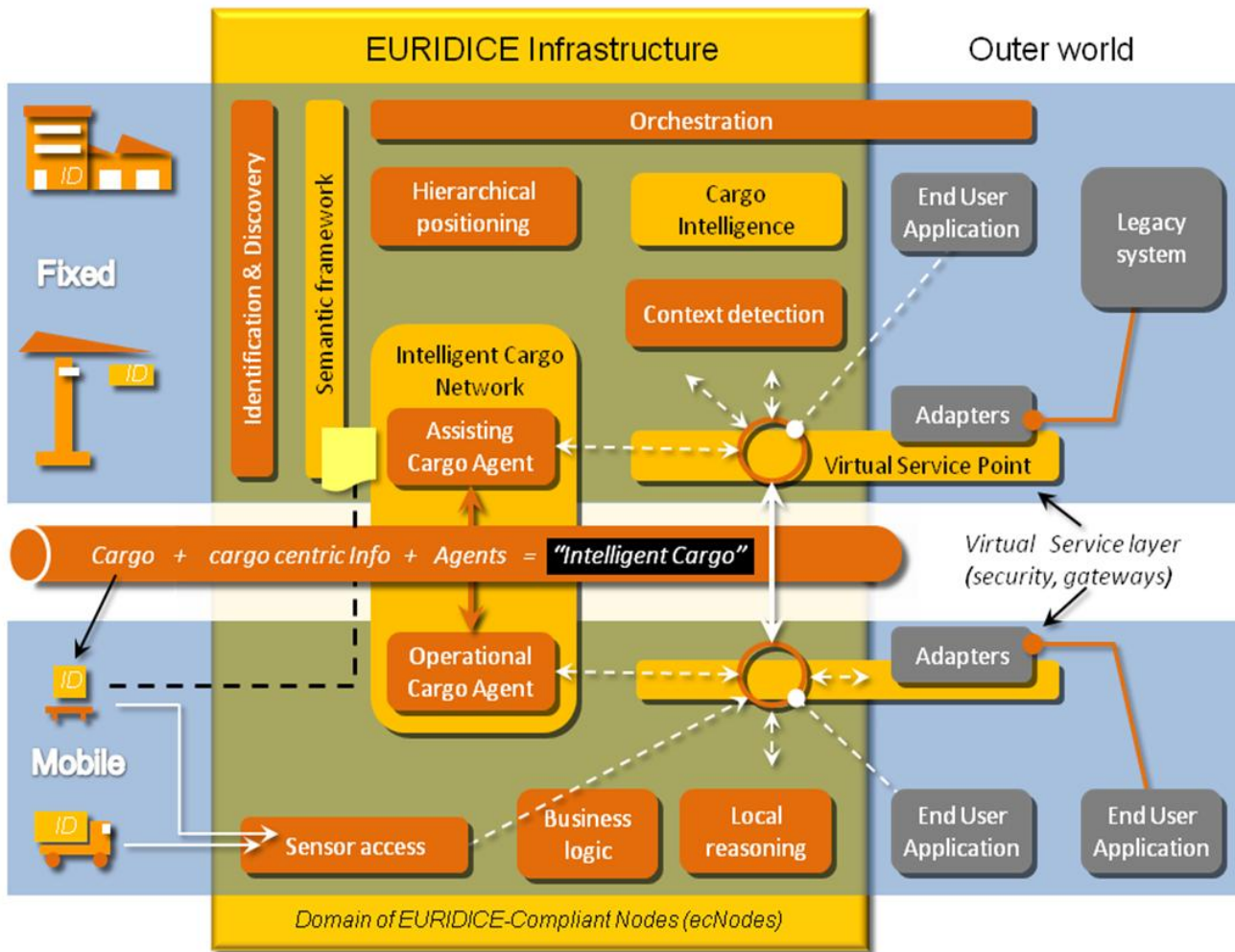


Figure 5: EURIDICE architecture and components

cargo item in the physical world by collecting and distributing information. The Operational Cargo Agent also includes local intelligence and can use it to:

- take proactive initiatives related to real-time information,
- provide information to interested parties
- make suggestions for measurements to take
- cooperation with other nearby intelligent cargo items, by sharing measurements for instance, as well as with the back-office

As implementation costs are still an issue, especially with low value goods, the functionalities can be delegated to a nearby mobile ecNode, e.g. container or onboard unit of a truck or train.

Virtual Service Point

This represents in the developed architecture a component regarding security issues. Information and services can be only accessed by a Virtual Service Point, enforcing security restrictions and providing secure communication services, to support the usage of one common platform between business entities and allows secure communication in a peer-to-peer like network. The Virtual Service Point routes authorized information and also requests for information to the required services. In this part of the EURIDICE architecture additionally also services will be hosted and implemented to allow the usage

of specific end-user-applications and integration with legacy systems.

Semantic Framework

For cooperation across organizations and business domains, a semantic framework, based on common data models and ontologies [6], reduces the barriers for adoption and communication between business entities. Therefore a semantic framework has been developed to support the usage and adoption of the core services of the platform, but additionally also business specific services, based on business related data models elaborated with the pilots in the project, supporting the development of end-user-applications and integration with legacy systems.

Context Detection

For providing the ability of interaction between cargo objects and systems regarding the environment of a cargo item, a common ontology and necessary data models are developed as part of the semantic framework, to enable services to detect and relate to the current context of cargo items. This provides a common understanding between EURIDICE services in the context of the cargo item, and additionally by usage of sensor measurements, simple "tracking & tracing" information can be enriched with additional dynamic accessible information related to the current cargo context, as well as the interacting entities,

e.g. freight forwarders, authorities and logistics services providers.

Cargo Intelligence

EURIDICE promotes as one of its key solutions the already mentioned Intelligent Cargo (IC) concept supported by global reasoning. Therefore a standardized knowledge structure (provided by an ontology) extended by rules, relations and actual measurements, trends and unusual deviations being determined are shared by the participating entities in the platform. This knowledge structure cannot be only used for centralized decision-making, but also for decision-making in the field, at the intelligent cargo itself for assisting other entities, e.g. human operators, in the decision-making process.

Orchestration

Orchestration relates to the same concept of a SOA, providing mechanisms for orchestration of services across business domains and to facilitate cooperation between stakeholders in the supply chain. This enables the combination of different services provided by participants in supply chains and users of the platform. Therefore highly specialized applications can be implemented based on single services or also by usage of sub processes.

SERVICES PLATFORM

The main objective of the project is to provide an information services platform with the focus on individual cargo items, their interactions with the surroundings and the stakeholders. EURIDICE therefore provides a fixed and mobile web services infrastructure, for enabling real-time access to cargo information, if needed, to private and public stakeholders along the transportation chain, supporting information retrieval related to the cargo for back-offices and field staff. The provided services can be divided into the following categories shown in Figure 6.

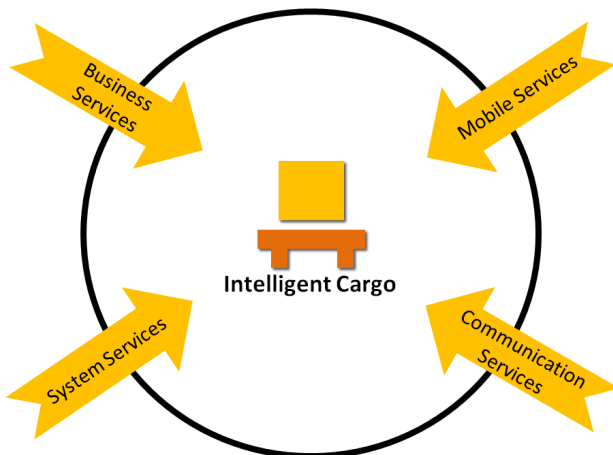


Figure 6: EURIDICE services platform

Business Services

These are intended as a set of fixed services for enabling integration with back-offices, 3rd parties and legacy systems. These services can be used for integration with existing systems, e.g. Enterprise Resource Planning (ERP), Inventory Management and Fleet Management.

Mobile Services

This is a collection of cargo centric mobile services. They are intended for the observation of the cargo, simple related to "tracking & tracing" and advanced features, e.g. sensing and anomaly detection. These services usually are running on mobile devices (intelligent devices), e.g. cargo items, containers, vehicles. Although these services are also available for cargo items only equipped with passive RFID-tags or barcode, with maybe less functionality.

Communication Services

Communication services are handling any kind of connection between users, vehicles and intelligent cargo items, as well the connection between intelligent cargo items and available sensors, e.g. RFID antennas, GPS receivers, temperature, humidity or any other available sensors available in the platform.

In a more technological view, these services are implemented by using a layering architecture related to the concepts of SOA, to reduce complexity in the system development phase itself and later to facilitate the implementation of business processes (Figure 7).

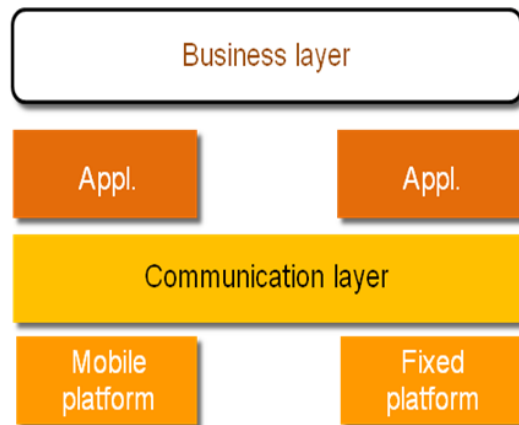


Figure 7: High level abstraction layering

Following layers are defined in a first high-level layering approach:

Business layer: In this layer agreements and procedures required for the usage and management of the application components as well as information to be exchanged are defined. This layer includes no software components, only necessary definitions.

Application layer: This layer contains the business services processing components.

Communication layer: This layer provides the necessary components for secure communication and information exchange independent form the underlying platform.

Platform layer: This layer is divided into mobile and fixed platform, as it represents the used hardware and the required operating system, if applicable.

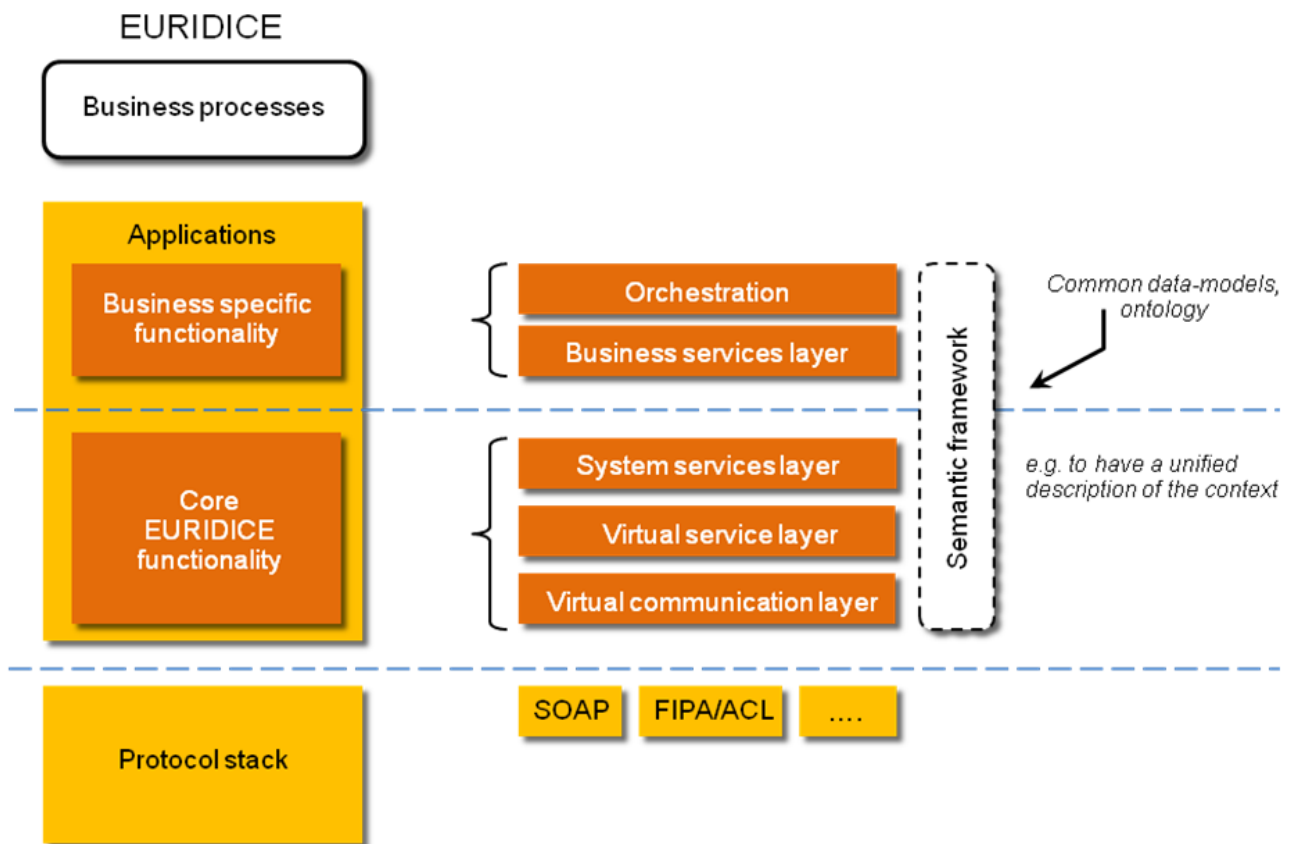


Figure 8: EURIDICE applications domain sub-layers

The application layer domain has been divided into several sub-layers for further reduction of complexity, as the main development will be performed in this domain (Figure 8).

The intention of the sub-layers is as follows:

Virtual communication layer: This layer is intended for decoupling of above services from a specific communication protocol, additionally gateways and system specific adapters are hosted here. One important component implemented in this layer will be the transformation from SOAP (Simple Object Access Protocol) to FIPA/ACL (Foundation for Intelligent Physical Agents / Agent Communication Language) and vice versa, exposing the available mobile agents (mobile services) implemented by available frameworks and standards as web services in a SOA.

Virtual service layer: This layer represents the single point of access to services and provided information including security and routing to applicable services after authorization.

System service layer: In this layer the EURIDICE core services are hosted, implementing the concepts of ecNodes and Intelligent Cargo Network, as well as other necessary system services provided by the platform.

Business service layer: This layer contains supply chain specific services providing also support for implementation of other necessary business specific services. Cargo Intelligence services, relying on actual input from other business services, are also hosted here.

Orchestration layer: This layer will contain the specific and necessary orchestration services for implementation of business processes. As already described and required by the concepts of SOA, the business processes will be implemented with a service orchestration language, in EURIDICE BPEL will be used.

The developed semantic framework is not visualized as a layer, but noted in the figure as a collection of common data-models and ontologies for a distinct interpretation of available information.

The EURIDICE architecture is positioned on top of the OSI (Open Systems Interconnection) [7] reference model and utilizes available communication infrastructures and communication protocols based on standards. A more detailed description of the EURIDICE platform architecture can be found in the official deliverables of the project.

The next part will deal in more detail with the distributed part and related technologies of the platform.

INTELLIGENT DEVICES

The miniaturization and the increased availability of on-board units allow the implementation of new business processes. The EURIDICE project takes advantages out of this new trend and extends the paradigm of ubiquitous computing by adding some context awareness to the application. Context awareness means that an application is able to sense its environment in terms of

detecting its global position, accessing several sensors or reacting on environmental events. [3] Not only transport vehicles, but also single cargo items can be equipped with quite sophisticated devices communicating with each other and other IT-systems. Every single part is responsible for a quite small amount of functionality where a combination of the single parts results in a quite complex and powerful system. If single nodes of the system need to communicate involving mobile devices, they need to use some wireless communication facilities like Bluetooth for short range and GPRS or UMTS for long range communication. The whole system becomes a distributed one which appears coherent and continuous to an end user. [8][9]

As physical and virtual parts of the EURIDICE platform are distributed to different components of the logistics infrastructure, different stocks, warehouses, vehicles, containers, single cargo items and additionally for the handling of the virtual processes, such a federative system (or such parts of a system) requires choosing of an architecture considering the limitations and challenges of a mobile computing system. [8]

Mobile devices work with limited resources. They have a limited power supply, limited CPU processing power and storage space. In most cases they do not guarantee a quality of service concerning the wireless connections and there are many more restrictions which have to be considered when developing mobile applications.

Therefore the platform implies more or less sophisticated devices (depending on the mission to fulfill) installed at single cargo items where in the best case the architecture can use the Multi-Agent approach with all its benefits. An intelligent, mobile agent is a piece of software that acts on the behalf of someone else, in our case on the behalf of a single cargo item where it represents its interests and goals. Agents work autonomously with absence of any user interaction. Their lifecycle is strongly coupled to the lifecycle of their representative. Agents work continuously and autonomously in a particular environment. [10] They communicate and cooperate with each other, with other processes and their environment (e.g.: via sensors) which makes them in some way intelligent and context aware. Agents usually act dynamically; they adapt their behavior based on the context they reside. The mobility of an agent does not only indicate that the devices hosting the software can be moved physically, it says that the agent itself is strongly decoupled from the hardware it runs on. Mobile agents are able to dynamically move from one machine to another. This approach is called 'migration' of an agent. [11] Such systems are able to be physically present almost anywhere and anytime which clearly seems to be very appropriate in the field of logistics where moving goods physically, from one location to another can be seen as the core competence of this business. Being physically present does not offer a lot of advantages on the first sight, but in a case where the need of real-time information, including environmental data measured by sensors (e.g. temperature via temperature sensor, position via GPS-sensor, availability of specific cargo-items in a container measured via RFID...) being physically present reduces communication problems, overhead, and time delays. The capabilities of the agents are offered as services and are the foundation for providing cargo-centric information services, observing the cargo and enabling an intelligent behavior for single cargo items.

Typical tasks performed by intelligent cargo agents could be tracking and tracing of goods, anomaly detection concerning the environment of the cargo in the meaning of critical temperatures, g-forces or humidity levels, but also wrong traveling directions, delays or early arrivals. The local computation approach minimizes the use of centralized systems to organize and redistribute the data.

By sensing the environment, the mobile and intelligent agents can react on changes or events appropriately by initiating counteractions or alarming a monitoring unit. An agent has the ability of adaption, as already mentioned, which means they are able to adapt their strategies to react on changes inside of the agent network and the environment by learning from their experiences.

The usage of a multi-agent architecture for this part of the EURIDICE platform becomes even more adequate, if we also consider design challenges apart from the enrichment of the responsibilities of cargo items within their processes:

Maintainability: Usually it is not possible to maintain distributed systems in traditional ways. The fact, that devices of massively distributed systems are not physically available to maintain them and deploy new versions of available software, forces different approaches to ensure a maintainable system. The multi-agent approach offers a way on one hand to create a highly customizable solution, which can be adapted dynamically at runtime, but also a highly maintainable solution, where software updates are realized using the code-mobility pattern of multi agent systems.

Scalability: Apparently software for the logistics sector must always set a high value on scalability. The diversified field of the logistics industry has to deal with an unforeseeable amount of items, where software systems need to be prepared for. In principal within a multi agent system an agent offers a quite small amount of functionality. Single agents use the capabilities of other agents to enrich their portfolio of expertise and to reach their goals and vice versa. Enlarging the functionality of the whole system means to simply introduce more and new types of agents.

Robustness: Robust software systems are systems that are prepared to handle failures and malfunctions. Especially in our case, where mobile agents physically accompany cargo, it is not possible to simply reboot a machine, if it fails, or restart a software component if it crashed. The devices used are not available for maintenance for a significant amount of time – until they reach the next stop of their shipment. It is not feasible to accept that in the meantime the system is not working anymore. Again the multi agent paradigm offers a solution to this problem. As it is the nature of multi agent systems to host cooperating agents, they also cooperate to increase the robustness of the system. This can lead to simple watchdog functionality, where agents observe each other, but also to more complex scenarios where redundant parts of the system are created dynamically at runtime, to overcome a blackout of single components. One possibility therefore could be the usage of code mobility, where a single agent clones and migrates to several other devices (devices attached to other cargo-items) and instantly hibernates there. The hibernated version of this agent will step in for the original one on failure.

Limitation of resources: Hosting software on single cargo items, containers or vehicles requires the availability of

hardware to host the software components. Where devices located on vehicles can assume in some way to have a more or less permanent power supply by being coupled to the vehicle's engine, devices installed on container or item level must rely on batteries and therefore can only use limited computation and communication resources. Beside specialized hardware delivered by several partners of the EURIDICE project to equip containers and vehicles, the system must also rely on very basic devices installed at single cargo items. Java enabled mobile phones offer computation capabilities and communication channels for both - long range (GSM, UMTS) and short range communication (Bluetooth). The architecture of EURIDICE is designed on one hand to offer the possibility to have fully featured devices on item level to host complex software, but also to offer mechanisms to deal with the problems of the limitation of resources. Mainly the concepts of the robustness and the key issues of the agent cooperation offer a way to share the anyway rare resources to avoid redundancy of tasks where not necessary.

Cargo cooperation: As mentioned before, one of the key aspects of a multi agent system is the interaction between agents and therefore cargo items. Cargo items interact for different purposes: they query other items for some information, they share knowledge and they consume services offered by other items. These services can be either be business specific services, as for example services for handling the business processes of single items, or observation of certain environmental parameters like the temperature or the position, but also infrastructure related services, e.g. long range communication channels, sharing of storage space or processing power. The pro-active cooperation between cargo items is one of the key aspects of the EURIDICE mobile platform. Not only the functionality of each single item is enriched, if it is able to use the functionality offered by other cargo items, also the robustness, the overall performance and the scalability benefit of the cooperation of single entities. The cooperation of items is especially used between items in proximity (e.g. items inside the same container). Equipping one device with a certain type of sensor (e.g. a GPS sensor) makes all items in proximity aware of their global position because the equipped item is willing to share its capabilities with other items.

Agent cooperation – Agent hierarchy

The issues already mentioned occurring in mobile computing applications need special attention because they are crucial. The energy consumption and the costs of communication of single cargo items are critical aspects of the overall architecture. Beside the communication costs, the potential lack of communication of mobile applications needs to be handled properly. Additionally a large number of mobile agents will be used in the project with the goal to observe the cargo, the business state and to offer real time information for business process optimization. Therefore within the EURIDICE architecture the usage of a cargo-hierarchy is envisioned – according to the physically occurring hierarchy of items. Single cargo items are placed within containers, where containers are transported by vehicles or stored within warehouses. Such a hierarchy can be seen in Figure 9. According to this hierarchy different responsibilities can be delegated to the single levels of the hierarchy where common services like positioning services or long range communication services are hosted on top levels and more specialized services, like the observation of the business process of a single item is hosted on the bottom of it.

a) Cargo Layer



b) Container Layer



c) Item Layer



Figure 9: Basic agent hierarchy in EURIDICE

The top level in the hierarchy (Figure 9 - a) is responsible for organizing the lower levels and to offer common generic services and communication facilities to the outside world. It represents the whole shipment for the rest of the system. The top level agent is installed on a quite sophisticated device, offering a more reliable power supply and worldwide communication services like GSM/GPRS or UMTS. It offers storage space to persist the data of the whole truck. Lower level agents typically use the common services of the upper ones. So they can be installed on more primitive and therefore cheaper devices.

The second level of the hierarchy (Figure 9 - b) is responsible for organizational issues within one container like item registration and de-registration, routing messages from the outside world to specific item agents, etc. It offers a yellow page catalogue for all devices and sensors within the container. This yellow page catalogue encourages the cooperation of agents in proximity. Agents entering the container register the services they offer at the yellow page catalogue where other agents can query the yellow page catalogue for services they need. The communication inside the container is mainly handled via short range wireless communication technologies like Bluetooth.

The third (Figure 9 - c) level consists of single cargo items. Single items can be equipped with quite primitive devices. They represent one single item, offer access to installed sensors on item level and host parts of the stateful business services for the concerning item. They can use the available resources of the whole agent network within one truck to extend their functionality.

Seamless integration of cargo items

We already discussed the integration of cargo items into the EURIDICE platform with the usage of mobile, intelligent agents hosted on devices installed directly at the cargo. But this approach is not feasible for all scenarios appearing in the logistics sector. There are cases where the hardware costs for hosting agents would exceed the transportation costs for a huge amount of cheap goods by far. Therefore only offering the integration of transported goods via the discussed approach does not seem to be sufficient to provide a pan-European platform for logistics services providers. For such cases the latest years have shown that RFID (Radio Frequency Identification) provides a cheap way to integrate physical items into software systems using the principles of the 'internet of

things'-paradigm. The mentioned approach strongly couples real world objects to the objects representing them in the virtual world. Today neither passive nor active RFID tags provide the capabilities to host such complex software components like mobile, intelligent agents. RFID tags usually show a passive behavior. With the help of RFID readers they can be activated which means to access and read the information they provide. This information can limit itself to simple identification purposes, but with semi-active RFID tags also more complex capabilities like measuring of temperature can be realized.

The EURIDICE architecture also offers a way to integrate cargo items only equipped with less sophisticated devices like RFID tags. From a higher level point of view, such cargo items show exactly the same behavior as items equipped with complex, more sophisticated devices. That means that basically all the services offered by EURIDICE-enabled cargo items are also available for RFID tagged items. To realize this transparent integration of different types of cargo, the architecture introduces the already mentioned concept of the Assisting Cargo Agent (ACA).

Assisting Cargo Agent – Operational Cargo Agent

Within EURIDICE a single cargo item is represented by a software agent [12] as already introduced in the section above. Cargo agents installed on devices in the field are called Operational Cargo Agent (OCA). Operational cargo agents always own a centralized counterpart – the assisting cargo agent. The assisting cargo agent – as its name already says – assists the operational one. It is deployed centrally within the EURIDICE centralized framework. Conceptually there is a one to one relation between an operational cargo agent and an assisting one. The assisting cargo agent shows the same interface and behavior as its operational counterpart. It is used for different purposes:

Central point of communication: The assisting cargo agent offers a communication channel for business services. It does not only offer a channel for direct communication with the operational agents – it also provides the same interface. So business services can directly consume services offered by operational agents using the interface provided by its assisting, centralized counterpart. Business services which need to interact with certain cargo items contact the regarding assisting cargo agent, which is available centrally. This concept strongly decouples centralized business services from the cargo items in field. They do not need to deal with communication issues and asynchronous interaction problems.

Support for poor equipped cargo: As already described above, the assisting cargo agent acts as a proxy for the operational one. Depending on the device used for the operational cargo agent, different services can be hosted on it. Fully featured devices are capable of hosting all kinds of services, where more limited devices do not offer the resources and capabilities to cover all the needed functionality. For that purpose the service portfolio for one specific item can be distributed between the assisting cargo agent and the operational one (Figure 10).

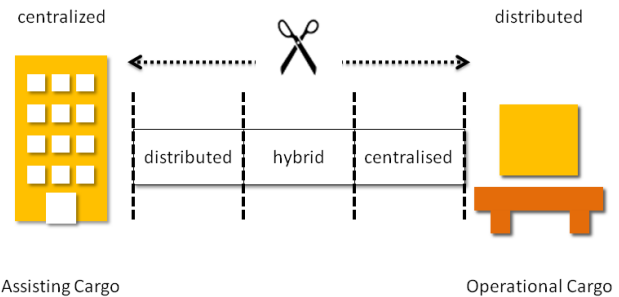


Figure 10: Functionality of Assisting and Operational Cargo

If one thinks of a band containing all functionality concerning a cargo, one can cut it into two pieces, where the left side of the band is hosted centrally and managed by the assisting cargo agent, and the right side is hosted remotely on mobile devices, handled by the operational one.

Depending on the type of device used for the operational cargo agent, the level of distribution respectively centralism can be configured. Cutting between point (1) and (2) would result in a very distributed approach where most of the functionality is hosted remotely by the operational cargo agent. Cutting between (3) and (4) shows the opposite behavior where a very small part of the functionality is offered by the operational cargo agent itself. Services that cannot be hosted remotely are covered centrally by the assisting cargo agent. Obviously this is not always possible or will lead to a bad quality of information. Observing the temperature of some cargo cannot be done centrally. For that purpose again the cooperation of agents can be used to solve such problems. Assisting cargo agents can delegate tasks to other cargo items – namely to other assisting cargo agents. They forward the tasks to their operational counterparts. That means that assisting cargo agents are able to cooperate with operational cargo agents from other cargo items which are in proximity of its own cargo. This opens the opportunity to also have high quality information gathered remotely for cargo that is only poorly equipped.

Decoupling components: The centralized business services, which live in a pure SOA world, must be decoupled from the underlying Multi Agent System. The business services use single cargo items as a direct source of information or they receive triggered events from them, driving their processes. Business services do not directly interact with operational cargo agents – they communicate with the assisting counterpart, which, from their point of view, directly represents the cargo item. A strong coupling between the business services and the operational cargo agents would not only lead to a hardly maintainable system, but also the behavior and the performance of the according web services would suffer. Where asynchronous communication and concurrent execution are central issues within multi agent systems, the consumption of a SOA based business services should appear transparently and reliable for end users. Assisting cargo agents handle requests from business services. Parts of the service portfolio offered by a cargo item can be hosted centrally inside the assisting cargo agent, where other parts are hosted distributed directly at the transportation good. If the assisting cargo agent cannot process a request itself, it forwards it to its operational counterpart.

Handle connection problems and enhance performance: As already discussed previously, mobile distributed applications always need to handle weak

connections. This means that programs cannot rely on permanent network connections. There are already mechanisms available to handle this unreliable, asynchronous communication between mobile applications. From the business services' point of view, the cargo itself is always reachable – via the assisting cargo agent. Again this concept decouples the SOA world from the distributed multi agent system. If the assisting cargo agent cannot establish a connection, it still has the opportunity to resolve the request via the resources available centrally. Scenarios like the usage of cached data or the cooperation with other assisting cargo agents can be used to handle such problems. A possible scenario could be where a business service requests the global position of an item. The assisting cargo agent first can check for some cached data. If it received the position from the operational cargo agent several minutes ago, this position can still be used to enhance the performance of the system. There is no real need to re-fetch the GPS data from the operational cargo agent. If no cached data is available, there is still the opportunity to ask other assisting cargo agents and their related operational cargo agents in proximity (e.g. inside the same container) available cargo items for it. In such a case cached data from items in proximity can also be used and saves unnecessary communication. If none of the mentioned strategies can be applied to overcome a connection problem, the assisting cargo agent needs to estimate and interpolate the position of the cargo for determination. Of course the quality of this estimation cannot compete in terms of accurateness with data fetched from an operational agent. Therefore generally, if a service requests information regarding environmental parameters measured by cargo agents it needs always to define a minimal quality for the information. This enables the assisting cargo agent to use the most economical strategy to respond to the request.

IMPLEMENTATION NOTES

For the implementation of the multi agent system architecture, an open source and FIPA compliant framework called JADE is used. It is published under LGPL and written in pure JAVA. Therefore it enables the use of quite cheap, JAVA enabled devices down to J2 Micro Edition – mobile-phones.

Software infrastructures and technologies

This part will give a short overview about the used software infrastructures and technologies for the mobile agent concept within EURIDICE.

FIPA (Foundation for Intelligent Physical Agents)

[13]: *“FIPA is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies.”* [14] FIPA defines some basic design issues, the services for the infrastructure and a communication standard between agents in a multi agent system. An FIPA standard compliant agent ensures interoperability to other systems utilizing FIPA. Moreover standardized libraries and middlewares can be used.

JADE [15]: *“JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications.”* [15]

JADE uses a FIPA-compliant agent platform which includes an Agent Management System (AMS), a Directory Facilitator (DF) and an Agent Communication Channel (ACC).

- The AMS is appropriate for the operations of the platform (e.g. creating an agent); it offers the description of the platform and is therefore the managing authority of it.
- The DF offers information to other agents about agents inside an agent network. Inside an agent network there can be several DFs with the possibility of forming federations.
- The ACC is a CORBA IIOP server object which listens for remote invocations.

JADE also offers some additional features like:

- Agent platforms can be distributed on different devices
- Connecting different agent platforms to a network
- Graphical user interface for managing agents and agent platforms

To run JADE on lightweight devices, which are running the J2ME-CLDC / MIDP 1.0 environment, the JADE-LEAP [16] library was developed.

CONCLUSION

The described platform offers an implementation of new logistics services that are strictly goods or “things” related, thus turning the current process orientated architectures of existing logistics systems around by placing the goods in to the middle of a SOA and the realized business processes on top of it. In principle this change can be compared with the change from functional to object orientated programming languages. This fundamental change will allow a much better usage and customization of services according to the real needs of users involved in logistics chains and offer new possibilities for adoption of innovative supply chain management concepts across organizational boundaries. By rearranging the service to concentrate on the individual cargo items, a new service environment for the logistics sector is created that allows the provision of innovative services on a coherent platform, leading to increased efficiency in the logistics sector and therefore supports the creation of environmental sustainability.

From a technical viewpoint the EURIDICE project combines the most advanced technologies available today to create a transparent and well-structured system that is able to handle the breathtaking complexity of modern logistics scenarios. Dividing the application into a more business-oriented, centralized part, hosted in a pure SOA environment and in a distributed part where single cargo items are included into the system with own virtual representatives hosted on mobile devices in the field offers various advantages. On one hand the possibility to control the strategic high-level processes via business related software like Business Process Engines and Service Oriented Orchestration and on the other hand it also covers most specific problem domains for single cargo items like sensing the temperature inside a truck. The presented solution maintains a clear and straight forward architecture and offers a scalable solution that is able to cope with the dimensions of today's logistics sector.

ACKNOWLEDGEMENT FOR PUBLICATIONS

This work was supported by the Seventh Framework Programme of the EUROPEAN Commission through the EURIDICE Integrated Project contract number ICT 2007-216271

REFERENCES

- [1] Directorate-General - Energy and Transport, European Commission. **Road Freight Transport Vademecum**, http://ec.europa.eu/transport/road/haulage/haulage_en.htm, 2009.
- [2] EURIDICE project website: <http://www.euridice-project.eu>.
- [3] W. Jia & W. Zhou, **Distributed Network Systems**, New York: Springer, 2004.
- [4] P. Braun, **Mobile Agents: Basic Concepts, Mobility Models, and the “Tracy” Toolkit**, San Francisco: Morgan Kaufmann, 2005.
- [5] EURIDICE Consortium, **The EURIDICE Project**. <http://www.euridice-project.eu/index.php/web/pubdocs>, 2009.
- [6] D. Fensel. **Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce**, Berlin: Springer, 2001.
- [7] OSI recommendation X.200: <http://www.itu.int/rec/T-REC-X.200-199407-I/en>
- [8] A. Greenfield, **Everyware – The dawning age of ubiquitous computing**, Berkeley: New Riders Publishing, 2006.
- [9] H. Attiya & J. Welch, **Distributed Computing: Fundamentals, Simulations and Advanced Topics** (2nd edition). Malden: John Wiley Interscience, 2004.
- [10] K. P. Sycara, **Multiagent systems**. AI Magazine, 19(2), 1998.
- [11] P. Braun & W. Rossak, **Mobile Agents: Basic Concepts, Mobility Models, and the Tracy Toolkit**. San Francisco: Morgan Kaufmann, 2004.
- [12] J. M. Bradshaw, **Software Agents**, Cambridge: The MIT Press, 1997.
- [13] Fipa.org, **FIPA Abstract Architecture Specification**, <http://www.fipa.org/specs/fipa00001/SC00001L.pdf>, 2003.
- [14] FIPA organization website: <http://www.fipa.org>.
- [15] JADE project website: <http://jade.tilab.com>.
- [16] Bellifemine & Poggi & Rimassa, **JADE – A White Paper**, <http://jade.tilab.com/papers/PAAM.pdf>, 2008.