Deliverable D13: LEAP Final Report

Lightweight Extensible Agent Platform

L E A P

Motorola
ADAC
Broadcom
British Telecommunications
Telecom Italia Lab
University of Parma
Siemens

Project reference: IST-1999-10211

Deliverable type: PP

Classification: Pub. / Int. (last section)

Work package and task: WP1, T1.3

Responsibility: Motorola

Executive summary:

This deliverable stands for the final report for project LEAP. It provides an update of the project objectives, methodologies and management aspects. The project main results in terms of technical achievements as well as dissemination, collaboration with other projects and impacts on standards are also evaluated to the initial objectives.

Annex A of this document consists in the annual report for the third period of the project: January 2002 to June 2002.

Contents

References

"Writing Effective Use Cases", Cockburn, A., Addison-Wesley, 2001.

"Methodology for Agent-Oriented Software Engineering", Evans, R., Kearney, P., Stark, J., Caire, G., Garijo, J.G., Jorge, J.G.S., Pavon, J., Leal, F., Chainho, P., Inovaçãi, P.T., Massonet., Peurescom project P907 , 2001. Available on-line at *www.eurescom.de/public/projectresults/P900-series/907ti1.asp*.

"The Rational Unified Process: An Introduction", P. Kruchten, Addison-Wesley, Second edition, March 2000

1. Project overview

Members of the LEAP project are associated through a consortium agreement, which has been signed at the beginning of the project. This consortium is composed of the six partner companies and one associated-partner university.

Motorola: is the co-ordinator of the project. As such it is responsible for the general management tasks and links with the European commission. Motorola has also a strong involvement in standards and had also a significant support in the technical activities. Motorola leads three work packages (WP1, 5 and 6).

ADAC: is the most important automobile club in Germany. ADAC employs 1,700 "Yellow Angels" for roadside assistance, who perform 9600 incidents / repair cases daily while driving about 150,000 km. ADAC represents the end-user point of view and experience in the project and hosts a field trial.

British Telecommunications (BT): is responsible for one work package aiming to develop the agent services (WP4), and also hosts a field trial: BT has 25,000 engineers performing 150,000 installation and repair tasks each day in the UK.

Broadcom: is an R&D centre jointly owned by Ericsson, Eircom and Trinity College Dublin. Broadcom is involved in all technical work packages.

Telecom Italia Lab (TILAB formerly CSELT): leads the technical work package, which aims at building the LEAP platform (WP3), due to their former experience with the JADE agent platform they have developed internally. TILAB is also involved in the development of the agent services.

University of Parma: is an associate partner of TILAB and has strong experience in agency through its Agent and Object Technology Lab (AOT). It is mainly involved in the technical development of the platform and had a very strong effort in dissemination.

Siemens: is the third device manufacturer to be part of the consortium. Siemens is involved in all technical activities and leads the work package dealing with the field trials (WP2). Siemens also supports ADAC in the preparation and execution of its field trial. In addition Siemens has chair position in FIPA standards.

Project LEAP is addressing the need for open infrastructures and services, which support dynamic and mobile enterprise workforces. It has developed agent-based services supporting three requirements of a mobile enterprise workforce:

- **Knowledge management** (anticipating individual knowledge requirements);
- **Decentralised work co-ordination** (empowering individuals, co-ordinating and trading jobs);
- **Travel management** (planning and coordinating individual travel needs).

Central to these agent-based services is the need for a standardised Agent Platform. Project LEAP has developed an agent platform that is:

- **Lightweight** (executable on small devices such as PDAs and mobile phones);
- **Extensible** (in size and functionality);
- **Operating-system agnostic** (written in Java);
- **Supporting agent communications over wireless links** (GPRS);
- **FIPA compliant**;
- **Released in Open-Source** (under the LGPL license).

The platform, along with the services have served in the development of two agent-based applications supporting mobile workforces, which have been assessed in two field trials, a first one hosted by ADAC in Germany and a second one run by BT in the UK.

2. Project objectives

Among its objectives, project LEAP distinguishes between the Lightweight Extensible Agent Platform and the agent-based services. The LEAP applications are the integration of both and are assessed via two field trials.

2.1. Goals of the Lightweight Extensible Agent Platform

As stated in Deliverable 3.1 *"Specifications of the LEAP architecture"*, this part of the project aimed to:

- Devise and implement a FIPA compliant Lightweight Extensible Agent Platform capable of deployment on both static devices (such as workstations and desktop computers) and mobile devices such as PDAs and phones;
- Demonstrate FIPA compliance in the form of inter-operation between the Lightweight Extensible Agent Platform and other FIPA compliant platforms;
- Realise an 'Operating System Agnostic' Lightweight Extensible Agent Platform, capable of deployment on several operating systems including: WIN CE, Palm OS, EPOC (provided that they support a Java Virtual Machine);
- Ensure that the Lightweight Extensible Agent Platform can operate in both wired and wireless environments over TCP/IP connections, so that the LEAP application can run seamlessly across both configurations;
- Manage the ability to configure the Lightweight Extensible Agent Platform with respect to the screen size, physical memory and other pertinent characteristics of a given target device. This configuration tool will manage the configuration of LEAP for several target platforms.

2.2. Goals of the LEAP agent-based services and applications

As stated in Deliverable 4.1 *"Agent services design"*, this part of the project aimed to:

- Prove that agent technology can add value in the management of mobile teams. Three kinds of problems are tackled: knowledge management, work co-ordination, and travel assistance;
- Demonstrate the advantage of locally based agents on small devices as part of a distributed application. Agents increase the autonomy of devices and applications;
- Prove the viability of both the infrastructure and agent-based services through two independent field trials deployed in Germany and the UK, running the same services, but in different operational contexts. Each field trial will involve users in vehicles, roaming over large areas, for a duration of one month;
- Demonstrate the use of a number of device/operating system combinations, during the field trials.

2.3. Description of work packages

Work package 1

Work package 1 activities lasted during the whole life of the project. Its objectives have been to co-ordinate and manage the project and to take care of the relationships and communication between project contractors and both the European commission and other projects. This WP was also responsible for defining the logistics of the project (selection of the tools, reporting structure and internal repository of software and documents); as well as controlling and refining the objectives of the project, and give assurance of timeliness and quality of project results and produced the quarterly and annual reports.

Work package 2

At the very beginning of the project, WP2 defined the requirement for the field trials (Task 2.1). Then this WP suspended its activities, while WP3 and WP4 developed the Lightweight Extensible Agent Platform and the LEAP applications respectively. WP2 resumed its activities in July 2001, to specify, and to prepare the field trials. In April and June 2002 the field trials were carried out. The last task of WP2 was the collection of field trial results and the evaluation of the field trials.

WP2s objective has been to demonstrate the business benefits of using distributed, heterogeneous software agents, in the context of mobile team working and to evaluate the operational performance of the LEAP application, when used in conjunction with the Lightweight Extensible Agent Platform and one or more target device/operating system combinations. WP2 had to produce a requirements analysis for the applications developed during the project, to carry out the preparation and the execution of the field trials as well as to produce the field trial result and evaluation report.

Work package 3

The objective of WP3 was to provide a Lightweight Extensible Agent Platform for supporting applications running on hand-held devices, over wire line and wireless communication links. More in details this platform had to be:

- Lightweight enough to be deployed on small devices, such as PDAs and mobile phones, with limited memory and processing power;
- Extensible so that, when deployed, on a powerful machine, it can provide a number of optional functionality such as agent mobility, user-defined ontology support and platform management GUIs;
- Operating system agnostic running both on Windows-like and UNIX-like systems and on handheld devices manufacturers proprietary operating systems;
- Transport layer independent and in particular supporting transport protocols suitable for both the wire line and wireless environment thus providing a homogeneous layer to agent application developers;
- Compliant to the latest FIPA specifications in order to be interoperable with other existing FIPA compliant platforms.

The ultimate goal of the WP is to promote the Lightweight Extensible Agent Platform as a reference FIPA platform around the world and to have it adopted by the largest possible community of users.

During the first period of the project, WP3 specified, designed and implemented a first version of the LEAP platform, by modify the existing JADE open source platform. The first version of LEAP was released in December 2000. Then WP3 concentrate on running LEAP on always-smaller devices, from laptops (Dec. 00) to PDA (Jan. 01) to phones (Apr. 01), besides reintegrating the LEAP libraries with the main development stream of JADE. The first version of the JADE-LEAP platform, based on the second version of the LEAP libraries and JADE 2.4 was released to the open-source community in September 2001. During the last period of the project, WP3 provided support to WP2 and 4 for the preparation of the field trials and also released a new version of the LEAP libraries (LEAP 2.1), to run with JADE 2.6.

Work package 4

The activities of WP4 were conducted in two phases:

- First, define and develop a set of generic services to support a mobile workforce. Different services have been identified and develop in each of the three families: travel management, knowledge management and teamwork coordination. The services are generic both in their customisability and their distribution.
- Then use this service framework to develop two field trial applications, by customizing the generic services to match the particular requirements of roadside assistance (ADAC) and network management (BT) and handle the specific information for each domain. These applications also include new technologies like GPRS, GPS, etc.
-

Work package 5

The activities of this work package run over the entire life of the project. Its objectives have been to assure the relation to standards (especially FIPA), to identify and participate in cluster activities and to disseminate the public results of the project outside the consortium via papers in conferences and the open-source activity.

Work package 6

The objective of Work Package 6 has been to describe and update the roadmap of the exploitation of results by the project contractors, especially regarding the open-source (LGPL) license. This WP was also responsible for evaluating the results of the field trials with a double focus: on the one hand it evaluated technically the LEAP platform, on the other hand it evaluated the applications by considering business, sociological and financial aspects.

3. Methodologies

3.1. The LEAP Approach: Development methodology

In order to achieve its challenging and innovative goals in a very changing environment, the project is conducted in two phases, each one following a similar cycle as depicted in Figure 1:

- **Phase 1** (duration 12 months), concentrates on feasibility and ends with lab trials running a skeleton application in a simulated environment.
- **Phase 2** (duration 18 months) starts from the lessons learned in phase 1 and leads LEAP towards the field trials running in the open air and covering large geographical areas.

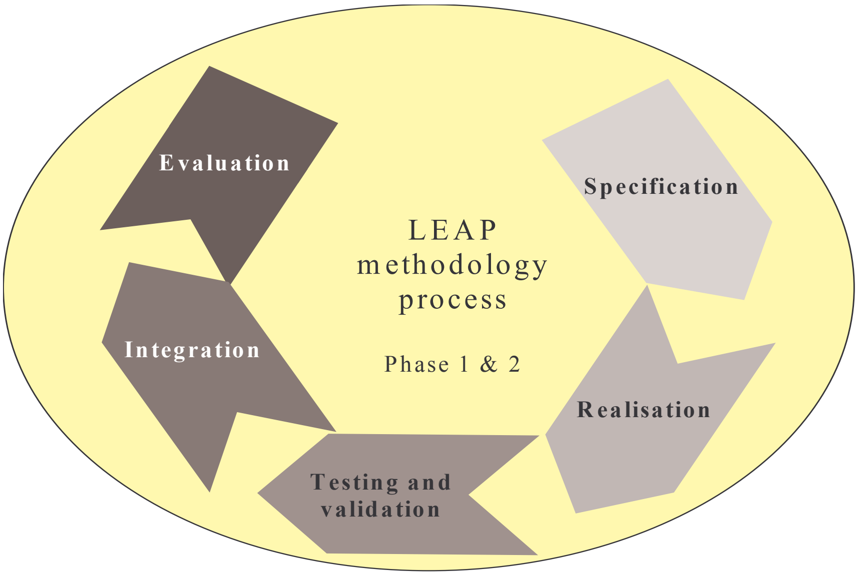

Figure 1: Overview of project phases

Each phase has the following fundamental steps:

Step 1- Specification

The project starts by defining the requirements for both the targeted technical achievements, as well as the requirements for the work environment, including:

- Software development environment;
- Communication infrastructure used (investigation of all possibilities including WAP);
- Test environments;
- Management of software policies and tools (like Control Version System);
- Specification method (such as UML);
- Methodology to co-ordinate software development across teams.

To be meaningful, these requirements are devised in close collaboration with people actually working in the application domain. At the same time, in order to promote both interoperability and an 'Operating System Agnostic' Agent Platform, the project reviews the FIPA '99 standard and studies the inter-operability trials carried out by FIPA.

In addition, the WAP protocol stack, WAP services and the WAP browser are closely examined, as to potentially support the Lightweight Extensible Agent Platform when it is running in the wireless environment.

Step 2 – Realisation
The design of the Lightweight Extensible Agent Platform and the LEAP services are carried out in parallel, and it leads to the implementation.
LEAP partners posses a wealth of relevant experience in building Agent Platforms, as five partners have already developed four different FIPA'97 compliant platforms, most of them being written in Java and running on workstations or desktops, using TCP/IP communications stacks.
The implementation of the Lightweight Extensible Agent Platform relies on the design and the development of a new architecture permits rightsizing, to suit the needs of each target device. Development will start with the kernel, create a modular architecture and develop its components - all other modules will follow the same development life cycle.
The development of the wireless communication stack (to sit alongside the existing wire line implementation) initially planned to re-use existing WAP stack, has been done from the own project experience. Development activities are distributed among the partners, using a shared code repository to exchange implemented and validated components. Repartition of tasks occurs during Technical Steering Committee meetings under the responsibility of the work package leader. Periodic implementation events, meeting up all partners take place to coordinate integration of components together.

Step 3 – Testing and validating
Due to the modular approach followed in the specification and realisation phases, all the developed modules are tested and validated separately in order to minimise the presence of unwanted features in the subsequent phases.
A particular emphasis is put on testing wireless communication between FIPA platforms. Context-independent tests are to be carried out in a simulated environment running on workstations or desktops, over TCP/IP, to prove the right use of the mechanisms in the Lightweight Extensible Agent Platform.

Step 4 – Integration
The Lightweight Extensible Agent Platform and the agent-based services are integrated and deployed in the simulated environment. The trials show how the skeleton of the agent-based services run, and assess both processing power and communications requirements. The trials are intended to prove the overall feasibility of project LEAP in simulated and real environments.

Step 5 - Evaluation of the trials
Each trial is followed by a qualitative and quantitative evaluation of the performance of the Lightweight Extensible Agent Platform, from which possible bottlenecks in both specification and realisation may be identified. The outcome of the lab trials will be used to drive relevant international standards, such as FIPA.

3.2. Other methodologies used

Agent methodology

Many MAS systems place the emphasis on a rich communication language to communicate *high level concepts* about information to distributed reasoning processes. The communication part in agent systems is used to some extent to co-ordinate and share information and services. It is the main way an agent externalises its requests or solutions to the rest of the community. Being able to communicate in a rich manner offers a potential infrastructure of openness, autonomy, robustness, scalability and flexibility. The trade off in order to reach these potential benefits is the complexity in maintaining coherence of distributed information through co-ordination. The main cost is how to co-ordinate these different services without placing too many restrictions on the communication language and the internal reasoning of the agent. There will inevitably be some restriction due to the modelling of complex application domains. The co-ordination feature of a MAS architecture is often distributed over four aspects of the communication language:

1. Protocols handled by the outer language (the content language is classed as an inner language). In knowledge query manipulation language (KQML) is partly supported by a performative and FIPA

ACL is supported by communicative act, FIPA protocol and linked to the content via an action definition;
2. Content part contains the expressions of encoded information and service details in a sharable way (via the language parameter and ontology model) for other agents to interpret syntactically;
3. Language used to express the syntax of the content expressions, e.g., Prolog;
4. Ontology model, which provides the explicit model of an application domain to allow an agent to apply an intended meaning to the content to be shared.

The actual reasoning about these aspects of a message, which is communicated from an external source (agent), needs to be internalised so the reasoning behaviour of an agent can deal with the intended meaning of the content. The receiving agent must take into account the protocol and its own belief model and current behaviour status of a set of current interactions.

Being able to communicate in a rich manner offers a potential infrastructure to support several key properties required by open distributed systems: for multi-agent systems the API is a communication language. The communication language defines certain protocols of computational execution. The communication language level of supporting computational interaction can, in some cases, be much "richer" than the classic representation of APIs in distributed object technology.

An agent as a bounded entity will use the classic notion of APIs, methods, predicates etc. to generate computational solutions as an internal process (AI would call this reasoning). To expose the rest of a computational system with the result of this internal process means using an agent type API.

The richness comes from the combination of protocol plus content. In affect the protocol is the method name and content is the method parameters, the receiver is the package handle that supports the method, the sender is the calling computational package. So by implementing many standard "classic" APIs you can get the same effect as would be achieved by using an ACL. Essentially you will have implemented an ACL from the bottom up approach. However, the abstract representation as an object abstraction is an a finer grained level which forces a developer to resort back to designing and implementing at a detailed level moving back to a more static development. For some problems it is better to design and develop at the higher protocol level hiding detailed development as an internal requirement. Working at a coarser level of abstraction provides potential solutions for:

1. Supporting openness of service provision and extension;
2. Deal with complexity in a particular service application;
3. Supports co-ordinating of service(s) in a distributed manner;
4. Supports distributed software development and integration.

The fundamentally difference between an agent approach and other engineering approaches is that when a message is dispatched from an agent, it is not object or method dependent but is a rich context of information that can be read (not necessarily understood) by another computational entity. A developer at the ACL level focuses on understanding a computational message protocol context. So part of the trade-off is whether development is more effective at the detailed API level or at interpreting the protocol context. We can assist in developing tools such as parsers and dialogue managers for the protocol context.

There are tools and parsers available to allow engineers to develop distributed software service systems through an agent approach. Developers can treat the communication language as a software interface. The interactions, dialogues or protocols are a set of calls to a particular program. Often an internal model of an agent interprets the input from a message, using both its current beliefs and the message to create a context in which to determine what to do next. Agent developers, like with most software development, will have a knowledge of both the expected input (communications received) and the expected output (communications sent). Protocols, communicative acts and content definitions are driven by the application requirements. Certain degree of separation can be achieved e.g. through the use of FIPA protocols to obtain a level of *architectural* openness.

Using an agent approach – the pros	Using an agent approach - the cons
Defines a set of key components for developing agent systems	Does not define patterns of composition, which would support agent development and assist interoperability, code mobility and code re-use at a service level.
Communicative acts for standard interaction	Currently most semantics is based on mental agency hence may limit interoperability
Protocols for complex message interaction have currently been defined	Not based on any semantics at this stage so interpretation between developers may differ
Definition of certain ontology requirements	Better support for ontologies is required if rich autonomous interoperability is to be supported for open services
Agent platforms increasingly support a variety of tools for managing agents, services and debugging.	Support for deadlock states, debug tools and general error handling is currently not specified - creating service agent platform dependencies
Flexibility is supported in the agent interaction requirements	Interaction lacks policies, e.g., agents are not compelled to answer hence termination of interaction may be difficult to establish.
There are a number of general reference models	There is not always a mapping between different reference models hence a certain degree of openness may be compromised.
Agent systems flexibly communicate using peer-peer interaction within a defined conversation protocol	Peer-peer interaction can create bootstrap and co-ordination problems. Conversation can prevent proactive behaviour by agents

Table 1: Summary Pros and Cons of using current agent systems (platforms, tools, languages etc.)

In designing agent systems, both by distribution of the engineers and the software itself when deployed, the approach used in the LEAP project utilised not just the abstraction protocol of agent communication but also:

1. The abstraction of task mapped to actions within the FIPA framework used by the communication protocols to achieve services e.g. such as swap task
2. The concept and definition of ontology for the communication of context in different tasks performed by the set of agents, which includes also the actions, supporting a better notion of ontologies compared with the current FIPA specifications
3. The separation of the service initiator and receiver, communication features, this addresses one of the cons provided in table 1 about service re-use, the development of service framework to assist in this abstraction and deployment.

Through the use of agent concepts, using a FIPA framework as a basis and addressing some of the support weaknesses for the design of agent-based services, the LEAP partners were able to develop the LEAP platform and generic service components, that where re-usable, interoperable and maintainable. The development and design if the LEAP platform also benefited from the JADE background development knowledge, use of UML design and component-based design. The details of the designs and knowledge design of the services was managed through the use of the message methodology. Further details of this approach is given in the next sections.

Use Cases

For the requirements analysis of the agent services use cases [Cockburn, 2001] were performed in each of the three mobile workforce domains (knowledge, travel, teamwork management). This resulted in 19 use cases, specifying goals, actors, triggers, pre and post conditions, success scenarios, possible extensions and open issues. The use cases are documented in D4.1.

Message

The MESSAGE [Evans, Kearny, et al, 2001] methodology was used to extend the requirements analysis into a high-level system design. The MESSAGE analysis model can be considered from five different views and at two different levels:

- **Level 0**: Considers the environment that the Multi Agent System (MAS) under development will be situated in.
- **Level 1**: Considers the internal structure of the MAS under development.

- **Organisation View**: This view considers the organisational aspects of the model. In particular it shows the various acquaintances between agents, roles and organisations, shows how organisations are be broken up into constituent parts and shows access to internal and external resources.
- **Goal/Task View**: This view considers goal decomposition and/or task decomposition. Tasks can be decomposed into work flows which can be used to implement some service. Understanding the relationships between tasks and how they can be chained together to implement services can be very instructive.
- **Agent/Role View**: This view describes in more detail the internal aspects of an agent/role as well as it immediate acquaintances, resources controlled and competencies.
- **Interaction View**: This view describes the interactions required by the MAS. During analysis we only record the net information flow and the parties involved in an interaction. We consider the actual interaction protocols in more detail during design.
- **Domain View**: The domain view is used to record the information entities required to describe the internal and external world of the MAS. This is typically done using OO techniques.

In addition the generalisation, aggregation and generic association relationships of UML can also be used. Three different types of diagrams were used in the analysis:

- **Organisational Diagram:** This diagram highlights the relationships between various parts of an organisation, e.g. organisations, roles, agents, resources, etc. A box is used to denote the limits of a system. Associations between these entities are as per UML except that:
  - Unnamed, undirected associations between agents, roles or organisations denote acquaintances.
  - Unnamed, directed associations denote responsibility towards a resource.
- **Workflow Structure Diagram:** This diagram highlights the implementation of a service through a partially ordered set of tasks. It consists of three components:
  - A parent pane at the top shows the entity and the service it provides
  - A child pane at the bottom shows those entities that are, or are associated with, the constituents of the parent.
  - A structure pane in the middle shows the partially ordering of tasks and events to implement the service. The directed associations between the tasks and events denote an ordering relationship. The tasks are also associated with entities in the child pane denoting capability, i.e. that a certain role/organisation/agent is capable of performing a task.
- **Interaction Diagram:** An interaction diagram denotes the information supplied, used and achieved during an interaction. It also denotes those entities which play the initiator and respondent roles during interactions.

In addition to these diagrams we further describe roles and tasks using a schema format.

- **Role Schema:** A role can be described by its identity, purpose, the requirements of an agent for playing this role, its interactions with the environment, its mental state and behaviour.
- **Task Schema:** A task can be identified by an informal description, which agent or role performs the task, the information that is inputted to and outputted from the task. Optionally pre- and post-conditions can also be attached to a task.

The software design tool Rational Rose was customised to display and work with the various notations within the MESSAGE methodology.

LEAP Generic Service Guidelines

Development of the generic services has been split between several partners, from different companies, located in different countries using the same code versioning tools as the development of the LEAP platform. Unlike the LEAP libraries, which follow the same development process, scheme and methodology as JADE, the LEAP generic services have been developed from scratch within the project and therefore guidelines needed to be defined in order to standardize the implementations produced by the different partners, ensure coherence and interoperability between generic services and ease the maintenance of generic service by any partner. These guidelines handle with: Initialisation, addition or removal of a service component, plugging and relationships with JADE-LEAP agents, service advertisement and lookup, utilization of service and format of the results, definition and use of timeouts, management of multiple service components within the same agent.

Rational Unified Process

The methodology described in [Kruchten, 2000] is very much inspired to the so-called Rational Unified Process (RUP). The RUP considers the development life cycle to be made up of a series of cycles called iterations where progressive product releases are developed. A software release is more than just the system code. The concept of software system includes all the artefacts that make it possible for the machines and the humans (workers and stakeholders) to interpret the description of the system. The term artefact refers to any kind of information created, produced changed, or used by workers in developing the system. Examples of artefacts are UML diagrams, user interface sketches, code libraries, and prototypes.

Each development cycle consists of four phases:

- **Inception phase** – Specifying the project vision;
- **Elaboration phase** – Planning the necessary activities and required resources; specifying the feature and designing the architecture;
- **Construction phase** – Building the product;
- **Transition phase** – Supplying the product to the user community.

Within each phase a number of iterations may occur. Iteration represents a complete development cycle where different activities such as requirement capture, analysis, design implementation and testing are carried out. The extent to which a particular activity is carried out is dependent upon the phase of development see Figure 2. For example the main activity during the Inception phase is Requirement capture, but it is also possible to do a proof of concept prototype and thus analysis, design, implementation and testing activities are also needed.

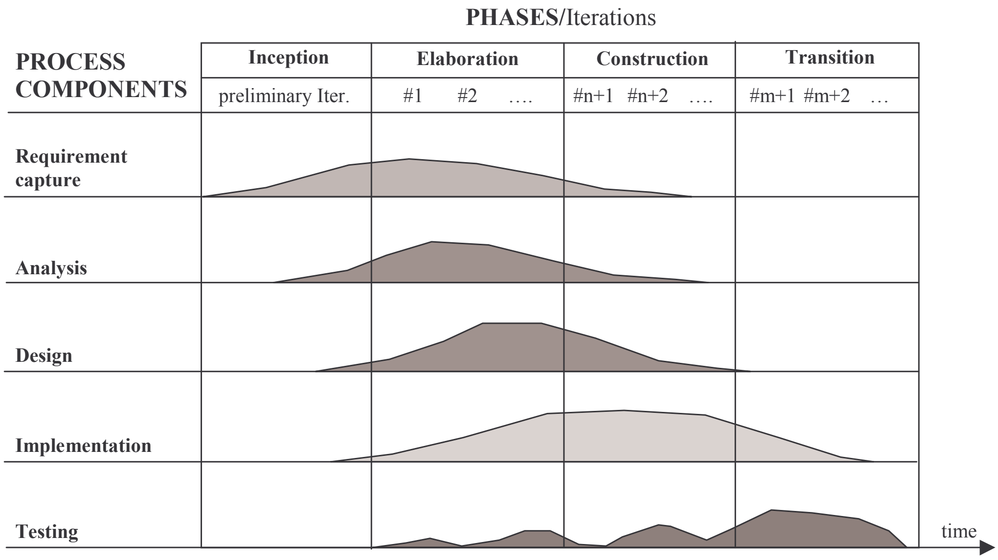

Figure 2. Phases, activities and iterations in the RUP

Test suite

As the LEAP libraries are distributed in open source they are accessible to a large number of developers. It is therefore very important that each new version is tested until a sufficient degree of robustness is reached before it is released. As the JADE-LEAP platform is quite a complex system supporting a lot of different functionality, testing all of them manually is a long process that can take several days.

In order to minimize the testing effort an automatic test suite was developed including proper test cases for all the main functionalities supported by the platform including (but not limited to)

- Agent startup/shutdown;
- Agent communication (intra-container, inter-container and intra-platform);
- AMS registration and search;
- DF registration and search;
- Agent mobility.

The test suite does not require any user intervention and, on completion, provides a report of all test cases successfully executed and indications about possible problems occurred during the execution.

Before each new release the following process is applied:

1) A new private (i.e. not accessible to the open source community) version of the LEAP libraries is generated;

2) This version is combined with the current distribution of JADE, thus obtaining JADE-LEAP;

3) The complete test suite is applied;

4) If all test cases are successfully passed the private version generated in 1) becomes the new official one and is made accessible to the open source community. On the other hand a new debugging session starts to solve the problems pointed out by the test suite report and the process is repeated.

The debugging sessions are carried out in tight collaboration with the JADE team as the bugs may reside in the LEAP part or in the JADE part or (as often happened) in some incompatibility between JADE and LEAP.

4. Project results and achievements

4.1. Lightweight Extensible Agent Platform

One of the most important achievements of the LEAP project was the realization of the “LEAP software”.
The LEAP software is composed of two main parts:

- The **LEAP libraries**: a set of libraries that, properly combined with the JADE platform (developed by TILAB outside the LEAP project), provide a runtime environment (called JADE-LEAP) that is compatible with Standard, Personal and Micro Java Editions (J2SE, pJava and J2ME-CLDC-MIDP).
- The **LEAP build tool**: an ANT-based tool that performs the process of combining the LEAP libraries with the JADE platform for a given target environment in an automatic way.

In order to promote JADE-LEAP as much as possible as a reference FIPA platform worldwide, the LEAP software has been distributed in open source under the LGPL license since September 2001 and can be downloaded from the LEAP site (http://leap.crm-paris.com).
The JADE-LEAP platform provides a homogeneous layer over a diversity of operating systems, hardware devices (ranging from servers to cell phones) and types of network and therefore allows ubiquitous deployment of agent applications both in the Internet and in the wireless environment (GPRS, UMTS, W-LAN).
JADE-LEAP is fully compliant to the latest FIPA specifications thus ensuring complete interoperability with agents running on other FIPA compliant platforms as proved for example in the Agentcities worldwide network of agent platform.

4.2. LEAP approach for mobile service development and deployment

Generic Service Components are the building blocks of applications for the LEAP platform. They can be organised into collections (referred to as ‘services’) and combined to support a large number of application scenarios. As the name implies, they require application-specific customisation in order to be deployed. This approach provides a number of reusable, discoverable components that can be used to make existing in-house, and third-party, service offerings available on a plug-and-play basis, as new services dynamically register with the platform’s directory service. Any combination of services can be delivered to many different device types – the chief requirement being that the device is capable of running Java.
Each of the three services identified to date (Travel Management, Teamwork Co-ordination, Knowledge Management) were analysed in turn using the MESSAGE methodology. The analysis was taken to design within document I41b.
Deliverable D43 provides a number of implementation guidelines (some mandatory, other optional) to assist developers in producing Generic Service Components for the LEAP platform. If the guidelines are followed, the resulting Generic Service Components should inter-operate seamlessly, and present standard APIs that allow other developers to readily customise them for application deployment purposes.
The knowledge gained from the practical experience of running a live field trial with an agent platform has been immense. We have learnt that although multi-agent systems may be just as good as other centralised systems for part of the mobile business process, certain key aspects of current and future mobile worker business processes tend towards the distributed nature of agent systems (e.g. soft rescheduling of jobs). Then, the agent system can be used as a platform for other more 'centralised' services. This results in a homogenous development and service distribution platform suitable for centralised and decentralised services to sit comfortably side-by-side on the same system - thus having the potential to reduce future integration development costs.
More importantly, the fact that the LEAP platform and services were capable (in terms of robustness, functionality, usability etc.) of being trialed in two separate operational environments has given the technology a gravitas that it would not otherwise have acquired at this stage in its development. This has lead to a number of opportunities for further development and deployment that are being actively pursued.

Therefore based on our trial experiences, we recommend that future technology projects include a trial element.

4.3. Implementation of generic services

This result consists in the Java source code and associated resources of the LEAP generic service framework, which is composed of 14 implemented generic service components. These service component are divided into three categories:

- Travel management services (e.g. plan-route, estimate-route-cost);
- Teamwork co-ordination services (e.g. give-task, make-collective-decision);
- Knowledge management services (e.g. update-knowledge-base, find-relevant-information).

An additional transverse family contains common services like user-authorization, brokering.
These service components are meant to be the building blocks for easily and quickly developing agent based applications. Generic service components need to be customized prior to use, in order to match the special needs of each application. They also have to be plugged within JADE-LEAP agents to run.
The overall framework comprises:

- Common interfaces;
- Code for the service components (initiator and respondent);
- Interaction capabilities;
- Action capabilities;
- The ontology for each family of service;
- Test examples.

A companion implementation guidelines document helps for the maintenance and the reuse of the generic services components.

4.4. Specification and preparation of field trials

During the work of WP2 a concrete definition of guidelines about how to prepare and execute Field Trials was developed and experience in preparing and executing field trials were collected.
The field trial preparation activities for both field trails were executed according to the developed definitions in D2.2 (specification of the field trials), e.g. build test versions, execute pre-field trial tests, select field trial team, prepare and execute training, check field trial entry criteria, equipment organisation. The decisions for the execution of the field trial made in D2.2 were adapted according to the current situation during preparation of the field trials (e.g. define final field trial dates and locations). Furthermore, evaluation criteria and different approaches for measuring the results of field trials are experienced.

4.5. Field trial applications and demonstrators

In order to achieve the project objectives, two trial applications were developed. Although similar in many respects, they supported different local business processes, interfaced to different back-end systems, and used two different locales (German and UK English). This result consists in these applications supporting ADAC and BT mobile workforces. They allow mobile workers to plan routes, calculate distances to tasks, retrieve relevant technical information on their task, find a convenient place to meet, swap their jobs, etc. Once the field trials had been completed, demonstrator versions of the two trial applications were produced, that were not reliant on access to computer systems running behind one or more corporate firewalls. This was done in order that each partner would have the capability to demonstrate LEAP both with their organisations, and to third parties as appropriate. This approach has also helped partners to build consensus for LEAP Technology.
<u>ADAC Field Trial</u>
The Field Trial was conducted in and around Munich, Germany and supported a group of Yellow Angels (mobile workers whom locate and repair the broken-down vehicles of members of the Allgemeiner Deutscher Automobile Club aka ‘ADAC’) during their normal course of work. The trial tested route planning, tourist information, technical information and meeting co-ordination services.
Two hardware configurations were used during the Field Trial in parallel:

- Siemens SX45 with GPS card and GPRS adapter.
- Compaq iPAQ 3870 with GPS card. GPRS connection was via Ericsson T39m mobile phone with Bluetooth connection.

BT Field Trial

The Field Trial was conducted in East Anglia, England and supported a group of Survey Officers (mobile workers whom plan) during their normal course of work. The trial tested route planning, just in time information management, flexible work scheduling and secure wireless networking services.

Two hardware configurations were used during the Field Trial in parallel:

- Compaq iPAQ 3870 with GPS card. GPRS connection was via Ericsson T39m mobile phone with Bluetooth connection.
- Compaq iPAQ 3870 with GPS card and GPRS adapter.

Device and service usage was logged throughout both trials, and detailed usage analysis was perform to determine the reliability, responsiveness and cost of using the LEAP services.

These applications are linked to ADAC and BT backend systems and access confidential information. For demonstration purposes, restricted versions of these applications, running over emulated data have been also produced. These demonstrators can be run without GPs information and over Wireless LAN connections.

4.6. Technical and social evaluation of field trials

During the project field trials in two domains were executed. One in the domain of telecommunication network maintenance (BT in UK) and one in the domain of roadside assistance (ADAC in Germany). Both field trials were running by using the LEAP applications and platform. For both field trials the evaluation results are summarized below:

ADAC: The ADAC trail application was designed to support ‘business as usual’ for Yellow Angels operating in and around Munich. The application provided route planning, tourist information, technical information and meeting co-ordination services. A mediator mechanism was used to pass messages through the firewall to agents on the mobile devices. In this scheme the LEAP servers also need to be granted access through the corporate firewall to access back-end systems.

The process of breakdown assistance was improved by enabling the yellow angel to get to the customer faster with the help of the route planning system, and to fix breakdowns faster by providing technical information. As there were some technical problems during the field trial no true ratio of improvement could be derived from the tests. But nevertheless it can be assumed that an improvement of about ten percent could be achieved with these functions. Above that, the LEAP platform showed to be able to be used as the basis for a mobile customer care centre to provide the ADAC member with information and additional offerings for e.g. insurance.

BT: The BT trial application was designed to support ‘business as usual’ for field Survey Officers (SOs) in the Eastern region of BT Retail’s UK Network Access Planning region. Each SO covers a unique geographical area or ‘patch’ and the SOs where chosen so that there were two notional teams each covering three adjacent patches to facilitate job sharing. Secure access to live BT servers was provided via an in-house wireless Virtual Private Network (VPN).

The BT trial application covered the following scenarios (see Figure 3):

- Urgent job notification (used as a trigger for ‘Give Job’);
- Give a job to a colleague (‘Give Job’);
- Plan a road route (Plan Route’);
- Retrieve plant and duct maps (‘Get Map’).

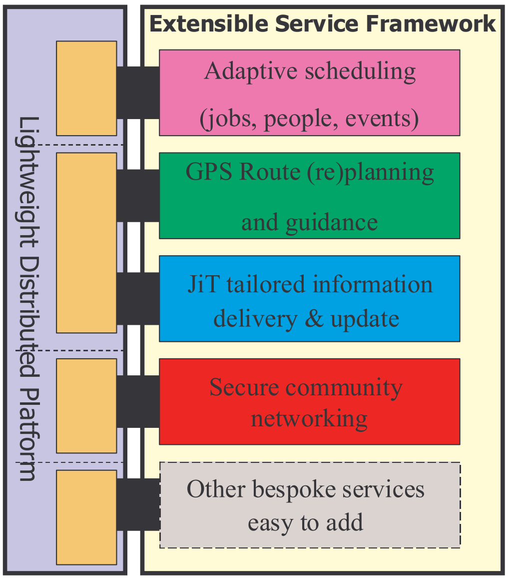

Figure 3: BT Trial Application – Service Mix

The trial applications actively demonstrated the effectiveness and usefulness of the LEAP approach to mobile workforces, especially in terms of service decomposition, service re-use, systems integration and the applicability of research orientated technology to real world business issues. The LEAP application is probably the first of its kind to demonstrate these benefits.

4.7. Impact on Standard

The project has made significant impact on the development of the FIPA standards through the following activities:

1. Developing and leading new technical committees: In particular, the TC Gateways, TC Ad hoc, X2S and ontology TC. These have been driven by both the technical and strategic activities of the LEAP project, especially with in the mobile workforce application space.
2. As well as the specifications and new work plans, such activities, as the test-suite, not only provided fundamental inputs to the X2S specifications, but also more fundamentally it has provided a way of conformance testing against the FIPA specifications. This has also meant that LEAP platform used frequently world-wide by many organisations to test if their platform is FIPA compliant. A clear and successful example of this is seen in the Agentcities.RTD project.
3. The development of some of the activities in LEAP has led to impacts on the Ontology developments. Some of this work is being submitted to the Ontology TC white paper for providing an ontological abstraction and support for services and application deployment. This work will also be submitted as part of the agent ontological requirements to the W3C.
4. Another related activity is the drive an input into the Agentcities task force, which will become an umbrella for impacting standards and providing recommendations in this space, the LEAP partners have and are participating to this activity to help with many standards convergence, such as, those current developments in the web-space.

The LEAP project and developments have had not only a broad impact on the FIPA standards, but also in the research activities in the agent community. The results are visible in such activities as conferences and workshops were the reference and use of the LEAP platform has been overwhelming (see papers published in http://autonomousagents.org/ubiquitousagents/).

4.8. FIPA compliance test-suite

This result consists in the FIPA compliance test suite designed and developed by the LEAP project. This test suite, which has been used internally to ensure FIPA compliance of the platform, is also now widely used worldwide for interoperability testing of FIPA platforms. For instance an implementation of the test suite, which runs on the JADE-LEAP platform of the Paris node, is becoming the de facto reference for FIPA-compliance of platforms in the Agentcities project. The specifications for this test-suite have also been submitted to the FIPA standard to become an official compliance test suite.

5. Deliverables and References

5.1. Project deliverables

No	Title	WP	Responsible	Type	Security	Delivery month
Technical deliverables						
D11	Selection of tools	WP1	Motorola	PP	Int.	3
D21	Requirements of Field Trials	WP2	Siemens	PP	Int.	4
D22	Specifications of field trials	WP2	Siemens	PP	Int.	20
D31	Specifications of the LEAP architecture	WP3	TILAB	PP	Int.	5
D32	Specification of the LEAP API + profile	WP3	TILAB	PU	IST	7
D33	LEAP version 1.0	WP3	TILAB	PP	Int.	12
D34	LEAP version 2.0	WP3	TILAB	PP	Int.	20
D41	Agent Services Design	WP4	BT	PP	Int.	8
D42	Lab trial	WP4	BT	PP	Int.	12
D43	LEAP Application Implementations	WP4	BT	PP	Int.	21
D62	Evaluation of Field Trials	WP6	Siemens	PU	Pub.	30
Deliverables & Annual review reports						
D12	Inter-working with other projects	WP1	Motorola	PU	FP5	3
D13	Project Summary	WP1	Motorola	PU	Pub.	30
D14	Annual Review report	WP1	Motorola	PU	Pub.	12
D15	Annual Review Report	WP1	Motorola	PU	Pub.	24
D51	Impact on standards	WP5	Motorola	PU	Pub.	15
D52	Impact on standards	WP5	Motorola	PU	Pub.	29
D61	Exploitation plan	WP6	Motorola	PU	Pub.	30
Additional deliverables (Not part of the initial technical annex)						
D12a	Inter-working with other projects (rev. 1)	WP1	Motorola	PU	FP5	3
D12b	Inter-working with other projects (rev 2)	WP1	Motorola	PU	FP5	3
D23	Results of Field Trials	WP2	Siemens	PP	Int.	4
D35	Maintenance activities	WP3	Motorola	PU	FP5	3
	LEAP and WAP positioning paper	WP3	TILAB	PU	FP5	3

Table 1: List of Deliverables

5.2. Overview of main deliverables

DELIVERABLE SUMMARY SHEETS

Deliverable: D21

Title: Field Trials Descriptions and Requirements

Short Description:

The mobile team management applications will be deployed in the real world, in two Field Trials, over a one-month time period, and covering large geographical areas in different countries. Both Field Trials are built upon generic agent-based services. They differ only in the specific business environment. In-the-van crews get orders, synchronise their tasks, retrieve documentation, interact with other team members to co-operate, get help and information and exchange tasks ‘on the fly’ according to their preferences and location. Pro-active agents running on their handheld devices will support them in all these operations.

This document serves to outline the purpose and scope of ADAC and BT Field Trials. It shows the common core of the applications and the adaptations to specific tasks. A short description of the Field Trials is given and possible applications of agent services are introduced with respect to tasks carried out by ADAC's and BT’s field workers/engineers. Requirement categories are identified for the system to use during Field Trials as well as to plan and execute a successful Field Trial of the LEAP project. These requirement categories are used for both Field Trials. The necessary requirements are based on these categories and criteria and will be defined in separate documents later (e.g. I2.1 and D2.2). Some examples of requirements are given in this deliverable.

Also, covered are all actions as well as documents to be produced in preparation as well as during the Field Trial in a so called "implementation plan". This implementation plan will be used as a guideline for the work of WP2 and finally for the preparation and execution of the Field Trials.

Partner responsible: Siemens

Deliverable: D22

Title: Specifications of Field Trials

Short Description:

In this document the detailed set-up of the LEAP field trials is specified, giving concrete information about dates, locations, and participants; the hardware, software, and networks deployed; and the organization of the field trials.

Partner responsible: Siemens

Deliverable: D23

Title: Field Trial Results

Short Description:

This deliverables summarises the results of the field trials at ADAC and BT in April and June 2002. In particular, it contains:

- A description of the final field trial system configuration at ADAC and BT (hardware, software, architecture, logging mechanism);
- A summary of the field trial process at ADAC and BT (dates, participants, actions);

- A collection of field trial results (analysis of log files, quantity of application and device usage, occurred problems and their statistics, e.g. network and battery, cost analysis, summary of debriefing results);
- A collection of concrete measurements during the field trials (evaluated log files represented as diagrams)

Partner responsible: Siemens

Deliverable: D31

Title: Specifications of the LEAP architecture

Short Description:

This deliverable addresses the analysis and the initial design of the LEAP agent platform architecture. An overview of the project LEAP is given outlining a number of key concepts, such as agent platform and agent application, and stating the high level requirements for the LEAP platform.

There are detailed analysis of the high level requirements with respect to the currently available technologies and provides a complete list of operational requirements. Also, described are the main architectural choices and, on the basis of these choices, analyses the main architectural elements. Included are the high level designs of the LEAP platform and a description of the installation and configuration facilities that will be provided with the platform introducing the concept of profile. Finally the report concludes with choices and illustrating that the design presented not only meets the requirements discussed, but also provides other valuable features.

Partner responsible: TILAB

Deliverable: D32

Title: Specification of the LEAP API and Profile

Short Description:

This deliverable provides the detailed specification of the APIs and profiles necessary for developing the core functionality of the first version of the LEAP platform. This platform is integrated with the JADE platform and hence expands its functionality to define an infrastructure, which enables the development of agent applications even for resource constraint devices, such as PDAs and mobile phones.

The overview and architectural details were provided in D31. The aim of this deliverable is a result of refining the architectural design and re-design of core JADE components. These interfaces, APIs and profile definitions permit the realisation of the software to meet the goals of D33. The first phase of the results of a working integrated system are detailed in deliverable D42.

This deliverable provides an overview of the convergence process of JADE and LEAP, the platform design, the application API realisation and profile definitions. The deliverable concludes with a summary of design decisions and to meet the constraints of the requirements.

Partner responsible: TILAB

Deliverable: D33

Title: LEAP Version 1

Short Description:

This deliverables presents the first version of the Lightweight Extensible Agent Platform, with source and J2SE compiled version released inside the package, and also guides how to use it. It describes the approach that has lead to LEAP implementation. Included is how to start a platform, add a container, and launch an agent. Explanations of message exchanges between agents are also provided.

Version 1.0 of the Lightweight Extensible Agent Platform (LEAP) was released in December 2000. This is the first FIPA 2000 compliant platform that allows deploying agents on small handheld devices (such as a Palm V) with a maximum of 200 Kbytes of memory for applications and supporting only a KVM compliant to the J2ME CLDC Specification.

The LEAP platform has been designed in order to be independent from the underlying transport protocol. This makes it possible to plug in new transport protocol implementations, even at run-time, provided that they are compliant to a well-defined interface.

LEAP therefore can provide a homogeneous layer spanning heterogeneous environments in terms of hardware and network connections both wireless and wire-line. Hence developers can focus on application requirements rather than development concerns.

These characteristics of LEAP paves the way for future m-commerce applications, where agents on PDA's and phones will provide Intelligent, Personal and Proactive Services to mobile users as well as to mobile teams. The LEAP platform however is not only extremely lightweight, but is also extensible so that, when deployed on a powerful device such as a PC, it also provides:

- A rich suite of graphical platform management tools;
- Additional advanced functionality such as agent mobility;
- A library of components supporting user defined ontologies.

Partner responsible: TILAB

Deliverable: D34

Title: LEAP version 2

Short Description:

This document presents the second version of the Lightweight Extensible Agent Platform, which has been released in Open-Source under the LGPL license on September 26th, 2001.
It briefly reminds the development choices that have lead to this version and introduces the content of the associated CDROM, which constitutes Deliverable D34.

This deliverable is a software package that includes version 2 of the LEAP libraries and the LEAP build tool and that is distributed in open source under the LGPL license.

This package can be used in conjunction with JADE version 2.5 to obtain JADE-LEAP, a FIPA compliant agent platform that can run seamlessly on servers and mobile phones and over the Internet and in the wireless environment.

With respect to the previous version the main improvements are listed below.

- A new version of the intra-platform communication protocol has been developed supporting full (logical) peer-to-peer communication over GPRS that is an intrinsically client-server network. This version is implemented by the JICPMPeer and Mediator classes included in the jade.imtp.leap.JICP package.

- The configuration parameters have been sensibly simplified so that now the user only needs to set very few and intuitive parameters to make the JADE_LEAP platform run.

- The build procedure (i.e. the process by means of which the LEAP library is merged with the JADE platform and the resulting JADE_LEAP platform is compiled for different target environments) has been simplified and made more robust. In particular the user does not need to set anything in order to build JADE_LEAP for the J2SE environment and only need to set two parameters in the buildLEAP.properties file to build JADE_LEAP for Personal Java and CLDC MIDP.

- The LEAP UserGuide has been sensibly improved: it is more readable and includes useful details especially about how to launch JADE_LEAP in a CLDC MIDP environment.

- New functionalities introduced in JADE2.5 are also made available in JADE_LEAP.

Partner responsible: TILAB

Deliverable: D41

Title: Agent Services Design

Short Description:

This document presents the top-level analysis and design of the agent services that form a crucial part of the LEAP application. The generic agent system modelling concepts are defined. The agent services followed by the ontology of the agent services are introduced. The services in terms of use cases and the interaction of each service that has existing legacy systems is described.

The agent system modelling concepts Agent, Goal, Task and Fact are first introduced, and their relationships to each other and their roles in agent systems are explained. An analysis of the concepts involved in the three agent services (travel management, teamwork co-ordination and knowledge management) is presented. These concept maps are used to provide the basis for the terminology of the agent services, which is defined in the agent services ontology. This terminology is then used to describe each component sub-service, which are detailed as Use Cases. Components from two or more of the agent services are combined to support the functional requirements of the LEAP Field Trials (one is concerned with rescuing stranded motorists, the other with installation and repair of telecommunications equipment). Finally, we summarise the requirements for the interaction of the LEAP agent system with legacy systems that arises from the use case analysis.

Partner responsible: BT

Deliverable: D42

Title: Lab Trial

Short Description:

The success of the Lab Trials has provided Project LEAP with: the confidence to go forward (as the platform was made to run a simulator at a relatively early stage in the project, and the laboratory application scenario on top of that, then the likelihood of successful implementation of the Field Trial applications is much greater); the first demonstrator with which to showcase LEAP, both within consortium member's own companies and to third parties.

Rather than member companies and third parties having to suspend their disbelief until near to the end of the project, the Lab Trials have provided tangible evidence that Project LEAP is viable.

Partner responsible: BT

Deliverable: D43

Title: LEAP Services and Applications implementations

Short Description:

Generic Service Components are the building blocks of applications for the LEAP platform. They can be organized into collections (referred to as ‘services’) and combined to support a large number of application scenarios. As the name implies, they require application-specific customization in order to be deployed.
This document provides a number of implementation guidelines (some mandatory, other optional) to assist developers in producing Generic Service Components for the LEAP platform. If the guidelines are followed, the resulting Generic Service Components should inter-operate seamlessly, and present

standard APIs that allow other developers to readily customise them for application deployment purposes.

Partner responsible: BT

Deliverable: D62

Title: Evaluation of Field Trials

Short Description:

This deliverables includes the evaluation of the results of the field trials at ADAC and BT in April and June 2002 in respect to the Balanced Score Card and the Annex I of the LEAP project proposal. In particular, it contains:

- An introduction to the Balanced Score Card to be used for evaluation (technological, customer, social and financial benefits);
- A summary of vision and project objectives from Technical Annex of the project proposal (already related to the fields of the Balanced Score Card)
- An evaluation of the results of ADAC and BT field trials (deliverable D2.3) according to the Balanced Score Card;
- Summary and Conclusions.

Partner responsible: Siemens

5.3. Main publications

Date	Title	Conference / Journal	Author(s)
Apr. 2000	LEAP – Lightweight Extensible Agent Platform	FIPA Inform! Issue 2	B. Burg (Motorola)
Oct. 2000	Towards the deployment of an open Agent World	Journées Francophones d'Intelligence Artificielle Distribuée et de Systèmes Multi-agent	B. Burg (Motorola)
Dec. 2000	LEAP: a Development and Run-Time Environment for Intelligent Agents	FIPA Inform! Issue 3	B. Burg (Motorola)
Jan. 2001	Mobile Virtual Management through the LEAP environment and services	FIPA Inform! Issue 4	B. Burg (Motorola) J. Shepherdson (BT)
Jan. 2001	Open Standards and Open Source for Agent-Based Systems	AgentLink Newsletter #6	B. Burg (Motorola) J. Dale (Fujitsu) S. Willmott (EPFL)
Apr. 2001	Agents LEAP onto lightweight PDAs	FIPA Inform! Issue #4	P. Charlton (Motorola) J. Shepherdson (BT)
May 2001	A Scalable Agent Infrastructure	Workshop on Infrastructure for Agents, MAS and Scalable MAS, Autonomous Agents'01, Montreal	M. Berger (Siemens) B. Bauer (Siemens) M. Watzke (Siemens)
Aug. 2001	LEAP: a FIPA Platform for Handheld and Mobile Devices	ATAL 2001, 8th international workshop on Agent Theories, Architectures and Language, Seattle	F. Bergenti (U. Parma) A. Poggi (U. Parma)
Sep. 2001	Enabling FIPA agents on small devices	CIA 2001 5th international workshop on Cooperative	G. Adorni (U. Parma) F. Bergenti (U. Parma)

		Information Agent, Modena, Italy	A. Poggi (U. Parma) G. Rimassa (U. Parma)
Sep. 2001	Porting Agents to Mobile devices	IEEE Internet Computing	F. Bergenti (U. Parma) B. Burg (Motorola) G. Caire (TILAB) A. Poggi (UParma)
Mar. 2002	Demonstrator Jukebots	CeBIT 2002, Hanover	M. Berger (Siemens) S. Rusitschka (Siem.)
June 2002	LEAP ADAC and BT applications	Siemens internal AgentDay, Munich	M.Berger (Siemens) D. Olpp (Siemens)
July 2002	Porting Distributed Agent-Middleware to Small Mobile Devices	Workshop "Ubiquitous Agents on Embedded, Wearable, and Mobile Devices", AAMAS2002, Bologna	M.Berger et al. (Siemens)
July 2002	A communication protocol for agents on lightweight devices	Workshop "Ubiquitous Agents on Embedded, Wearable, and Mobile Devices", AAMAS2002, Bologna	G. Caire (TILAB) N. Lhuillier (Motorola) G. Rimassa (U. Parma)
To appear (Sept. 02)	Ubiquitous Information Agents	International Journal on Cooperative Information Systems	F. Bergenti (U. Parma) A. Poggi (U. Parma)
To appear (end 02)	LEAP: A FIPA Platform for Handheld and Mobile Devices	Intelligent Agents VIII	F. Bergenti (U. Parma) A. Poggi (U. Parma)
To appear (end 02)	What Agent Middleware Can (And Should) Do For You.	Applied Artificial Intelligence Journal	A. Poggi (U. Parma) G. Rimassa (U. Parma) P. Turci (U. Parma)

6. Project management and co-ordination aspects

The general aim of the project management task is to ensure that all objectives are achieved on time and to budget. In case of any deviation from the plan, appropriate corrective actions will be determined and implemented. The project management structure employed in LEAP is shown in Figure 4.

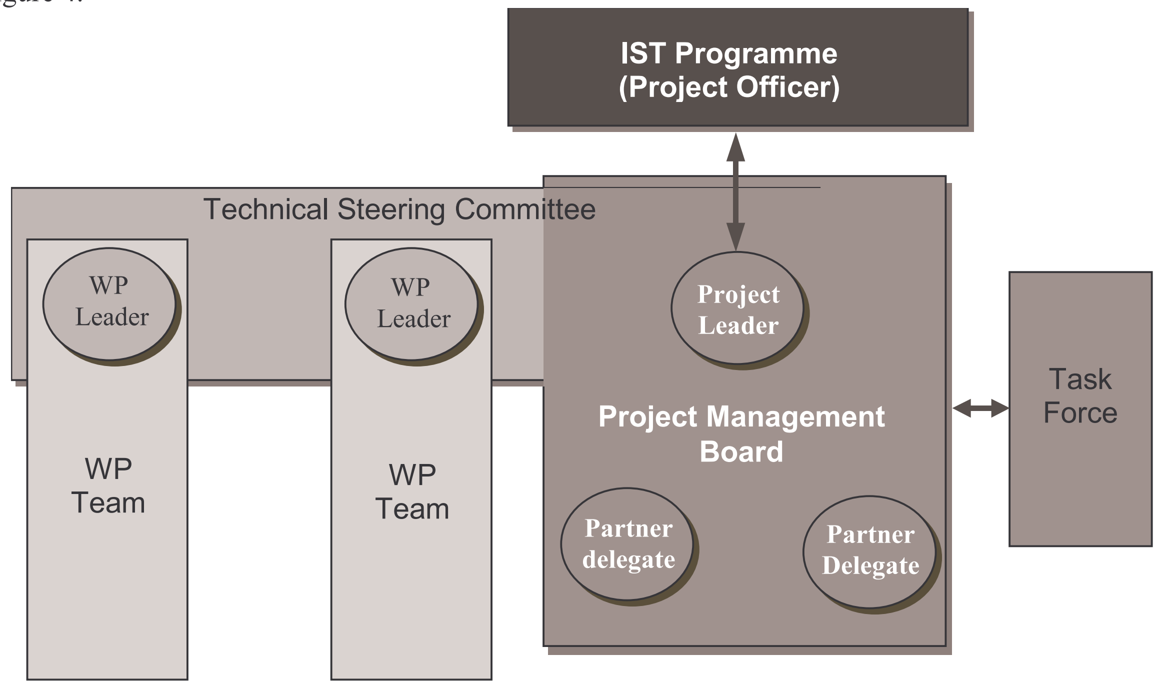

Figure 4: Project Management Structure

The **Project Leader (or Project Co-ordinator)**, acting as the representative of the prime contractor, is the permanent reference point of project LEAP on a day-to-day basis vis-à-vis both the contractors and the European Commission. The project co-ordinator is in charge of the overall project organisation, planning and reporting.

The project co-ordinator shall take care of:

- The relationships with the IST Central Office (including overall Project funding);
- The organisation and chairing of the Project Management Board and Technical Steering Committee meetings (see below);
- The dissemination of project results;
- The monthly and yearly reports;
- The overall project presentation to the audit meetings;
- The delivery of the cost statements.

The Project Management Board (PMB) consists of an official delegate from each full partner and is chaired by the project co-ordinator. Associated partners are represented by the relevant full partner.

The PMB is in charge of the high level management of the project, addressing all the administrative, contractual and financial matters. It takes all the important decisions related to the contractual execution such as:

- Contract changes or re-negotiation;
- Change of consortium configuration;
- Reallocation of works, responsibilities and man-power between contractors;
- Settlement of problems or differences between contractors.

The PMB is also responsible for defining the positioning of the project with respect to the activities of the Standardisation and Specification bodies, relevant projects in the same area and companies active in the field. Consequently, the PMB determines the strategic direction of the project.

From a technical point of view, the project is broken down into a number of **Work Packages** (WP) each of them addressing a specific area of work. The practical work of a WP team shall be conducted by contributions to periodic meetings, and by partial contributions, which will be submitted by individual members for formal adopted by the group. Each WP will be co-ordinated by a **WP leader** (see table below).

WP	**WP1**	**WP2**	**WP3**	**WP4**	**WP5**	**WP6**
Leader	Motorola	Siemens	TILAB	BT	Motorola	Motorola

The technical co-ordination among WPs is handled by the **Technical Steering Committee** (TSC), which is composed of all WP leaders and is chaired by the project co-ordinator.
In particular the TSC is responsible for:

- The implementation of the directives of the PMB;
- The guidance and monitoring of the technical work packages and the co-ordination among WPs;
- The timely preparation, approval and forwarding to the Commission of the Deliverables produced by the WPs;
- The resolution of conflicts amongst WPs.

If necessary, the PMB creates ad-hoc **Task Forces**, composed of experts, chosen from the project participants, that will work together to solve well-defined problems in a limited period of time.

Meetings and Decision Process
The PMB and TSC meet every three months in order to check and supervise the progress of the project. Decisions of the PMB and TSC are made by a simple majority of full members of the consortium (if not explicitly stated otherwise), each full member having one vote. In most cases decision have tried to reach a consensus among the partners.

Active WPs hold meetings at least every three months, in conjunction with PMB meetings. Depending on the work to be carried out intermediary meetings are organised in order to solve time-critical issues or monitoring progress of activities. To minimise travel costs, such meetings make use of audio conferencing facilities when possible. Two kinds of WP meetings are foreseen:

- Planning meetings, where all relevant decisions about the work to be carried out within the WP will be taken
- Integration meetings, where WP members will sit down in front of terminals to integrate and finalise their contributions.

Planning and Reporting
The project co-ordinator is responsible for the entire project in terms of organisation and administration. This responsibility covers the overall planning and reporting of Project LEAP. The WP leaders are in charge of planning and controlling the quality of the work and deliverables produced by their WP.

Reporting
Three kinds of reports are provided to the European Commission: the Monthly Control Report sent by group of 3 at the end of each quarter, the Annual Project Review Reports at the end of the first two years and the Final Report at the end of the project.
For the reporting purpose, LEAP created EXCEL reporting workbooks, which include planned and spent resources information and a description of the accomplished tasks, which helps in the preparation of the reports to the commission. These reporting workbooks are now used in other European projects.

Miscellaneous management activities

Types of main activities:

- Consortium agreement: the members of the project signed a consortium agreement, which was approved by each partner;
- The partners decided to release the LEAP platform in open-source. This uncommon non-commercial activity required approval from each company involved in the project;
- Due to internal turnover, both project coordinator and technical coordinator have changed during the second year of the project;
- A company has changed name: CSELT became TILAB;
- The technical development of the project were expected to highly rely on WAP technology, but WAP was indeed not available and failed to achieve the expectations so a workaround solution had to be found;
- For several months prior to the field trials, BT had been developing scenarios based on the day to day working of BT Retail Access Network installation and repair engineers. This involved shadowing engineers in the field, working closely with process design experts and producing and agreeing an internal document that described the scenarios in some detail. However, due to an internal re-organization, BT Retail withdrew their support for a trial in January 2002. That meant we had until April to find a new trial host within BT, repeat the process for developing scenarios and implement the BT trial application. In the event, the application was ready in time, thanks to the LEAP Generic Service Components, which had been designed and built independently of any of the trial applications, and required only customisation to support the new scenarios;
- An exploitation agreement has been signed between the partners in order to clarify the clauses of the consortium agreement and allow an optimal exploitation of the results of the project by each partner.

Conclusions

Project LEAP achieved all its objectives and also exceeded some:

- The LEAP software, properly combined with the JADE agent platform, allows deploying ubiquitous agent-based applications on a wide range of Java-enabled devices, which run seamlessly on wire line and wireless networks. JADE-LEAP additionally provides compliance with FIPA specifications and interoperability with other FIPA platforms, and also access to all functionalities provided by JADE such as support for content languages and ontologies, support for interaction protocols and a rich suit of graphic management and monitoring tools.
- The LEAP software developed and released in open-source has already been downloaded by more than 1 000 people, who are using it for their projects. The modular design of LEAP allows external contributor to add additional features and to make them available to the community. The number of papers referencing LEAP in the 2002 conferences attests the already important popularity of the platform. LEAP members continue to provide support to these developers and users through the leap-info mailing list.
- The field trials have been the first step towards new kinds of connected applications that support mobile workforces. The integrated use of new technologies such as intelligent agents, GPRS, GPS, though the risk and difficulty it includes, provides new pieces of functionality that will significantly change the way mobile workers currently operate, giving them technical support in their daily tasks, increasing the interactions with their co-workers, giving them more responsibility and more freedom. The fact that these applications are receiving a great interest during demonstrations is very promising.
- Project LEAP also drove the FIPA standards towards mobile devices. The Gateways and, newly created, Ad-hoc Technical Committee pave the way towards new kinds of agents on lightweight devices and new kind of collaborations due to the characteristics of these devices, in which the JADE-LEAP platform is expected to play a major role.

Appendix A. Periodic Progress Report (Year 2002)

This section is not public