INFORMATION SOCIETIES TECHNOLOGY
(IST)
PROGRAMME

Contract for:

Shared-cost RTD

Annex 1 - "Description of Work"

Project acronym: ***TAPAS***

Project full title: ***Trusted and QoS-Aware Provision of Application Services***

Contract no.: *(to be completed by Commission)*

Related to other Contract no.: *(to be completed by Commission)*

Date of preparation of Annex 1: **4 January 2002**

Proposal number***: IST-2001-34069***

Operative commencement date of contract: *(to be completed by Commission)*

Contents

1. Project Summary

1.1 Objectives

The overall objective of the TAPAS project is to develop novel methods, tools, algorithms and protocols that support the construction and provisioning of Internet application services. The project will achieve the overall objective by developing QoS enabled middleware services capable of meeting Service Level Agreements (SLAs) between application services and will enhance component based middleware technologies such that components can be deployed and interact across organisational boundaries. The project will develop notations for expressing SLAs to enable specification of QoS, such as the availability as well as trust relationships. SLA trust specifications will be used for deriving service invocation primitives enriched with authentication, non-repudiation mechanisms, with or without the involvement of trusted third parties.

1.2 Description of the work

TAPAS is structured into four technical workpackages (WPs). WP1 will develop notations for expressing SLAs to enable specification of QoS as well as trust relationships. Model checking capabilities will be developed to support reasoning about QoS characteristics of components and their composition. The project will adopt UML as the language for the description, modelling and analysis and extend it with formally defined stereotypes and properties. SLAs are only useful if their compliance is enforced and monitored. To achieve this aim TAPAS will use SLAs not only as an inter-organisational contractual feature but also to govern component execution. WP2 will develop support architectures that provide QoS negotiation, establishment and adaptation facilities to enable component containers to become QoS enabled. SLA trust specifications will be used for deriving service invocation primitives enriched with authentication, non-repudiation mechanisms, with or without the involvement of trusted third parties. WP3 will implement key middleware services identified in WP2. Particular attention will be placed on the development of QoS enabled multi-party communication (e.g. for supporting publish/subscribe communication, dynamic load balancing between replicated containers). Distributed applications with multiple senders pose several problems for network QoS; novel signalling methods and protocols for end to end QoS negotiation for resource reservation and feedback will be developed. Middleware services will be developed using open source application servers and widely used component technologies such as CORBA and JAVA. WP4 will evaluate results from the TAPAS projects by comparing them with current state-of-the-practice (e.g. an off-the-shelf CORBA or J2EE application server). Partners will build demonstrator applications with demanding QoS requirements such as hosting of an auction service.

1.3 Milestones and Expected Results

The final report on TAPAS architecture will cover the architecture of component oriented middleware for deployment in inter-organisation setting. Results will include model checking tool to support reasoning about QoS characteristics of components and their composition prototype, implementations using open source application servers and demonstrators. Preliminary results (milestones) will include reports on application hosting and QoS networking requirements.

2. Project objective(s)

2.1 Overall goals

Organisations, particularly small and medium scale enterprises (SMEs), are increasingly finding it difficult to develop, maintain and manage their IT applications largely due to difficulties in retaining and attracting trained IT staff. *Application Service Providers* (ASPs) hold the promise of providing an attractive solution by making available application hosting facilities on remotely managed servers. However, to work effectively, ASPs must guarantee security, provide resilience and *service level agreements* (SLAs) over commonly available infrastructures. Furthermore, ASPs need to ensure that hosted applications are capable of accessing a wide variety of services irrespective of the platform or the organisation through which they are provided. An ASP typically uses middleware and component technologies for deploying, hosting and managing applications of an organization from a centrally managed facility. However, as organisations become global and distributed, such centrally managed hosting solutions will need to be replaced by multi-site, distributed hosting solutions.

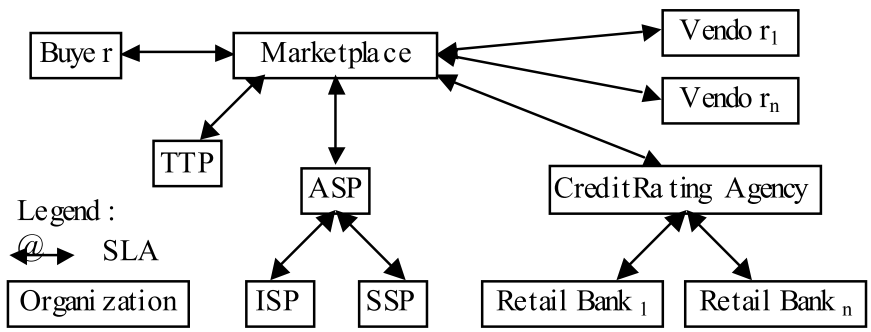
The goal of TAPAS is to develop multi-site, distributed hosting solutions. TAPAS will contribute to the community's social objectives by bringing improvements in several areas, including but not limited to: the quality of life; employment; enforcement of ethical values, such as privacy. Indeed, the project addresses directly some aspects that are crucial for the expected development of the Information Society, the achievement of which is one of the objectives of the Community.

We argue that many research problems in enterprise distributed computing will need to be solved to achieve the goal of TAPAS. An ASP will increasingly be called upon to host distributed applications that make use of a wide variety of Internet services provided by different organisations. This naturally leads to the ASP acting as an intermediary for interactions for information sharing that cross organisational boundaries. However, despite the requirement to share information and services, autonomy and privacy requirements of organisations must not be compromised. Organisation will therefore require their interactions with other organisations to be strictly controlled and policed. This creates two major challenges. Firstly, contractual relationships between multiple organisations for information access and sharing will need to be governed by SLAs, which will need to be defined and agreed between the organisations and then enforced and monitored by the ASP. Secondly, the ASP will have to establish appropriate *trust relationships* with the organisations and implement corresponding security policies before organisations will permit the ASP to act as an intermediary for inter-organisational service invocations. Unfortunately, middleware services for inter-organisational interactions as outlined above do not yet exist; indeed, development of such services is very much a research problem. Thus ASPs currently lack tools and techniques for offering hosting facilities for advanced Internet based applications.

2.2 Main objectives of the project

The main objective of the TAPAS project is to develop novel methods, tools, algorithms and protocols that support the construction and provisioning of Internet application services.

We use a simple example to focus on application hosting issues and limitations of current technologies. Our example considers the hosting of a Marketplace application that matches buyers and vendors. We might imagine that the Marketplace provides services through a component that manages requests for proposals (RFP). This component might use a service from a credit rating agency in order to obtain a credit rating of the buyer that is then forwarded alongside the RFP to vendors. The credit rating agency itself may implement its services using data obtained from account history services provided by retail banks. The Marketplace may need the services of one or more trusted third parties (TTPs) to meet the security requirements of buyers and vendors. We assume that the ASP hosting the Marketplace is obtaining its communication and storage resources from an ISP (Internet Service Provider) and an SSP (Storage Service Provider) respectively. The figure below shows the various organisations and the corresponding SLAs involved. The company providing the Marketplace has set up SLAs with buyers and vendors etc. and to meet the obligations implied by these SLAs, it will have a comprehensive SLA with the ASP specifying overall processing, storage and communication requirements, as well as availability responsiveness and security requirements.



In effect, to host the Marketplace, the ASP will require a distributed execution environment with a number of core services capable of meeting specific non-functional requirements of fault tolerance, availability, security, and timeliness; we will refer to these as *QoS enabled services* (QoS: Quality of Service). State-of-the-art application services are developed using component-based technologies, such as those provided by the Java 2 Enterprise Edition (J2EE), Microsoft's .NET or the Object Management Group's CORBA Component Model. These technologies support the cost effective creation of services by composing and integrating existing components. To date components can be obtained from component vendors through component catalogues and be deployed in-house. In the future, as our example illustrates, components that implement services will be hosted by vendors or dedicated service providers in application servers and will be invoked from components from other organisations across public networks. Current component technologies support the specification of functional component interfaces. They, however, do not adequately support the definition of the non-functional characteristics of component execution.

2.3 Operational Goals

TAPAS will achieve the main objective by developing QoS enabled middleware services that will enable components to be deployed and interact across organisational boundaries. We have translated the main objective into three operational goals which we describe below together with our approach to achieving them.

2.3.1 Goals related to SLA specification, Service Composition and Analysis Techniques

The TAPAS project will develop notations for expressing SLAs to enable specification of QoS, such as the availability, performance and scalability characteristics of components, as well as trust relationships. Model checking capabilities will be developed to support reasoning about QoS characteristics of components and their composition. This will support the ASP in assessing the qualities of service perceived by end-users prior to developing and deploying an application service. In our example, the composition of complex application services from more elementary services provided by different organizations will thus be governed by multiple SLAs. The project intends to adopt UML as the language for the description, modelling and analysis and extend it with formally defined stereotypes and properties. This approach is motivated by the large number of users of the UML in industry, who will find it easier to adopt a small UML extension than an entirely new specification language.

2.3.2 Goals related to Trusted and QoS-aware Services for Application Hosting

Component oriented middleware promotes the use of *containers* to host component instances. Containers are responsible for using the underlying middleware services for communication, persistence, transactions, security and so forth. TAPAS will develop support architectures that provide QoS negotiation, establishment, and adaptation facilities to such services; these will be used by containers to make them QoS enabled. Particular attention will be placed on the development of QoS enabled multi-party communication (e.g., for supporting publish/subscribe communication, dynamic load balancing between replicated containers) that cross organisational boundaries. To this end, SLA trust specifications will be used for deriving service invocation primitives enriched with authentication, non-repudiation mechanisms, with or without the involvement of TTPs.

2.3.3 Goals related to Assessment

The results from the TAPAS projects, in the form of methods, tools and techniques for design and development of component oriented middleware for provisioning of Internet application services will be evaluated by comparing the results with current state-of-the-practice (e.g. an off-the-shelf CORBA or J2EE application server). The TAPAS consortium includes an ASP, who will provide requirements for QoS enabled application service provision, and case studies. Partners will build demonstrator applications, such as hosting of an auction service to demonstrate the effectiveness of TAPAS results.

2.4 Evaluation of Results

The deliverables of the project by means of which we provide measurable evidences of the extent to which we succeed in meeting the above stated very challenging objectives will include prototype implementations using open source application servers and demonstrators as well as technical reports and papers.

We intend to build a particularly demanding application (in terms of QoS) that ASPs are currently unable to host using present day middleware platforms. The application we have chosen is that of electronic auctions. We will be able to demonstrate features such as the use of QoS enabled group communication for supporting timely delivery of notifications from the auctioneer to bidders, and supporting replication of auction servers for availability and so forth.

The inclusion of an ASP in the consortium will speed up the migration of TAPAS results and technologies to industry. In addition, the project will form an Industry Advisory Board, whose membership will represent a cross-section of technology providers and end-users; the Board will help us in revising, where necessary, the objectives of the project. Thus, even though the TAPAS project has been formed to address the needs of ASPs, the results of the project will be of interest to much wider scientific and industrial communities.

3. Participant list

List of Participants

Participant Role	Participant number	Participant name	Participant short name	Country	Status*	Date enter project	Date exit project
CO	1	University of Newcastle upon Tyne	UNEW	UK	C	1	36
CR	2	Adesso AG	Adesso	D	P	1	36
CR	3	Universita di Bologna	UniBo	I	P	1	36
CR	4	University College London	UCL	UK	P	1	36
CR	5	University of Cambridge - Computer Laboratory	UCAM-CLAB	UK	P	1	36

*C = Co-ordinator (or use C-F and C-S if financial and scientific co-ordinator roles are separate)

P - Principal contractor

A - Assistant contractor

4. Contribution to programme/key action objectives

The TAPAS project is specifically addressing most of the main objectives of *V.1.12 CPA 12, the Cross-Programme Action on Application Services provision*: to develop and validate open architectures, technologies and tools to allow for the provision of a variety of applications as networked services, with focus on development of middleware services and development of service management frameworks.

The TAPAS project will develop QoS enabled middleware services capable of meeting SLAs agreed between application services, and will enhance component based middleware technologies such that components can be deployed and interact across organisational boundaries. Middleware services and architectures will be developed using open source application servers and widely used component technologies such as CORBA and Java.

Broadly speaking, the TAPAS project is addressing two priority areas identified in *section 2.3 of the 2001 Work Programme,* namely:

- To develop middleware, distributed systems, multi-layered architectures and agents-based systems to enable interoperability, inter-working, openness and integration of application and services across platforms.
- To emphasise trust and security, including information security, privacy, suppliers and users rights and dependability of systems and infrastructures, as a general requirement for all technologies, applications and services.

The above priority areas map onto the following main *action lines* that are related to TAPAS:

- *II.2.2 Smart Organisations*, with focus on "development and validation of novel architectures, software platforms and pre-standards to support interoperability, seamless integration and knowledge sharing between heterogeneous enterprise applications and services".
- *II.4.1 Trust in information infrastructures*, with focus on "development, integration and validation of trust and security technologies in information infrastructures"
- *IV.2.1 Real time distributed systems*, with the objectives "to develop and assess models, technologies and tools for sharing and interactive use in real time of applications and resources in geographically dispersed locations…", with focus on "development environments to support real time, distributed applications".
- *IV.3.1 Software architectures*, with focus on "models and notations for describing systems architectures and being able to reason about them. The main concern is to guarantee required quality attributes (for instance on scalability, performance and reliability) of systems."
- *IV.6.2 Interfaces and buffers for seamless end to end services*, with focus on "network programming interfaces".

5. Innovation

We restate the two observations that underpin the TAPAS project objectives for application service provisioning, namely: (i) contractual relationships between multiple organisations for information access and sharing will need to be governed by SLAs, which will need to be defined and agreed between the organisations and then enforced and monitored by the ASP; and (ii) the ASP will have to establish appropriate trust relationships with the organisations and implement corresponding security policies before organisations will permit the ASP to act as an intermediary for inter-organisational service invocations. The TAPAS aim of developing QoS enabled middleware services capable of meeting SLAs and supporting components that can be deployed and interact across organisational boundaries poses significant technical challenges. We expect that the project will introduce several research innovations in the areas of architecture, system design and validation. We begin by highlighting the innovative features of the project:

- Currently, SLAs are mainly used for specifying network QoS requirements. In TAPAS we will extend the use of SLAs across all levels of service usage and develop specification techniques and analysis tools that will support an ASP in assessing the qualities of service perceived by end-users prior to developing and deploying application services.

- TAPAS architecture will use SLAs not only as an inter-organizational contractual feature but also to govern component execution. To this end, QoS negotiation, establishment, and adaptation facilities will be added to the middleware and these will be used by component containers to make them QoS aware.

- To enable inter-organisation interactions, SLAs will be enriched with trust specifications and used for deriving service invocation primitives with authentication, non-repudiation mechanisms, with or without the involvement of trusted third parties.

- In this project, we will consider hosting of important class of large scale networked *multi-party applications* that are characterised by a group of entities requiring many-to-many interactions; examples include collaborative applications (e.g., multimedia conferencing, multi-person interactive games), electronic share dealing and auctions, and 'information supply chain' applications typically based on publish-subscribe paradigm.

- Particular attention will be placed on the development of QoS enabled multi-party communication to support the above class of applications. For such applications we will concentrate on QoS properties of fault tolerance, availability, and timeliness. Currently, strict separation exists between the middleware that executes application services and the network, so providing services that go beyond 'the best effort' is difficult. TAPAS will examine ways of 'opening the network cloud' and develop network QoS API for multi-party traffic control.

5.1 SLA specification, Service Composition and Analysis Techniques

Ideally, an ASP should have at his disposal a toolkit that enables him to describe the configuration of the application to be hosted in terms of building block services, and then use the toolkit to confirm that the various SLA commitments can be honoured for a variety of physical configurations and load conditions. This should enable the ASP to make an informed choice between various design options *before* commencing the actual deployment of the application. Current system development practices fail to live up to this ideal; rather, systems are frequently built without any precise idea about how the system will actually perform. The reason being that there is no rigorous methodology available to system designers that allows them to express their design in sufficient detail, and predict what the system behaviour will be with respect to, say, timeliness under a given load. The TAPAS project aims to overcome this deficiency by defining a language for SLAs. Alongside functional components interfaces, these SLAs specify the non-functional qualities of service, such as the availability, performance and scalability characteristics of components. The project intends to adopt UML [1] as the language for the description, modelling and analysis and extend it with formally defined stereotypes and properties. TAPAS will define the semantics of SLAs using stochastic process algebras [2-6]. Model checking capabilities developed for stochastic process algebras will then support reasoning about the performance and scalability characteristics of components and their composition. This will support the application service provider in assessing the qualities of service perceived by end-users prior to developing and deploying an application service. currently SLAs are used mainly within the rather narrow confines of network service performance, TAPAS will extend SLAs to application level services.

5.2 Trusted and QoS-aware Services for Application Hosting

QoS-aware services: SLAs are only useful if their compliance is enforced and monitored. To achieve this aim, TAPAS will use SLAs not only as an inter-organizational contractual feature but also to govern component execution that adheres to specific QoS requirements as implied by the SLA (commonly referred to as *QoS policy* of the application). In order to provide QoS-aware execution of components, application servers themselves need to rely on a number of basic middleware services. The TAPAS project will aim to define the distributed algorithms (QoS signalling protocols) and implement the middleware services that are necessary for QoS enabled deployment and execution of components. To achieve this aim, TAPAS will extend open source application servers (such as Jboss [7] or Jonas [8]) to enforce and monitor adherence to SLAs.

So far, issues of QoS have been addressed principally in the design of communication protocols. In this context, the principal design concern has been that of providing mechanisms that allow one to master and control quantitative communication parameters, such as network throughput, delay, delay jitter, and packet loss. These parameters indeed affect the quality of the service perceived at the application level. However, it is at this very level that further QoS application requirements emerge; these may include performance oriented requirements (e.g. timeliness of execution and relative processing speed requirements), reliability oriented requirements (e.g. high availability and relative failure recovery requirements), and security oriented requirements (e.g. authentication, privacy). There is thus a need for research on QoS enabled middleware services that takes a broader view of QoS than has been done so far. TAPAS will develop techniques for translating the QoS policy of an application into low level resource requirements as required by QoS signalling protocols. This is a relatively new area of research, particularly for multi-party applications (see next).

Network level QoS for multi-party applications: The Internet community has made two efforts to develop standards for Internet Protocol (IP) with QoS capabilities: the integrated services architecture (IntServ) and the differentiated services architecture (DiffServ). There is much ongoing debate concerning strengths and weaknesses of these two architectures which we will not discuss here, but both expose the need for sophisticated mechanisms for accessing network resources [9,10]. In addition, because of the large impact that multicast can have on a network, the design of multicast management middleware (for services such as creation and joining of multicast groups, multicast address management, and providing various forms of reliability guarantees) is quite complicated and design issues not properly understood. Distributed applications with multiple senders pose several additional problems for Network QoS. Traffic engineering techniques currently used by network service providers for route balancing and prioritising traffic need to be extended to the situation involving multiple senders; this will require studying traffic patterns from these applications (looking for correlation in the traffic from different sources that make up an application). An important question for the network QoS API is ‘how to encode efficiently a multi-party traffic specification’; clearly, specifying the full matrix of source models is not going to be viable, but the alternatives are not obvious. The other side of this coin is that a multi-party application can have a partial QoS contract failure. This needs to be signalled efficiently to the relevant parties in the application by the network; multicasting techniques for such signalling needs investigation.

Trusted coordination: To enable inter-organisation interactions, SLAs on service usage, defining what is to be shared, who is allowed to share, costs etc will need formalising. The organisations involved might not trust each other, so an important part of our research will be concerned with developing infrastructure support for interactions between two or more mutually distrusting organisations. This is a relatively new area of research [10-14]. The relationship between parties may be asymmetric (customer-supplier) or symmetric (peer-to-peer). Parties may be strangers to each other or there may be a pre-existing basis for trust. Trust between parties may develop as a result of the interaction. Trust relationships may be symmetric - mutually trusting or distrusting - or asymmetric, when trust (or distrust) is not reciprocated. Parties engaged in an interaction require an infrastructure that both facilitates the interaction (for example, to share and coordinate application state) and provides guarantees with respect to protection of their individual and collective interests, given the trust relationships. So, if neither p or q trust each other then interaction between them will need to be coordinated via one or more mutually trusted third parties. For trust management, TAPAS will develop non-repudiation and authentication services as well as robust fair exchange protocols [15-19]. We will also need to develop notations for specifying trust relationships, so that from that specification, 'right' kinds of infrastructure components can be derived.

6. Community added value and contribution to EU policies.

The following quotations from Council Decision on the Fifth Framework Programme indicate the importance to Europe of the issue of achieving interoperability of systems:

"It will not be possible to realise the full potential of the information society in Europe with only today's technologies and applications. Key requirements such as usability, dependability, interoperability and, above all, affordability are far from being sufficiently met for the broad deployment of information society technologies (i.e. information and communication technologies, systems, applications and services) in all areas"

"Realising the full potential of the information society requires technologies, infrastructures, applications and services, accessible and usable by anyone, anywhere, anytime, whether it be for business or individual use. Collaborative research and technological development is needed to create both the critical efforts and the interoperability necessary to ensure this in Europe"

The IST Workprogramme itself is very clear about the need to achieve interoperability of systems:

"The aim of the IST Programme is to help to create a user-friendly information society by building a global knowledge, media and computing space which is universally and seamlessly accessible to ALL through interoperable, dependable and affordable products and services."

In particular, the 2001 Work Programme has identified two priority areas, namely:

- To develop middleware, distributed systems, multi-layered architectures and agents-based systems to enable interoperability, inter-working, openness and integration of application and services across platforms.
- To emphasise trust and security, including information security, privacy, suppliers and users rights and dependability of systems and infrastructures, as a general requirement for all technologies, applications and services.

These sentiments are of course among the major motivations for the creation of CPA 12, *the Cross-Programme Action on Application Services provision*. The action line description of CPA 12 states as its objective:

"To develop and validate open architectures, technologies and tools to allow for the provision of a variety of applications as networked services over a commonly available infrastructure; to develop methods for the management and dynamic allocation of computing, storage and communication resources and for the monitoring of usage and service quality by providers and users; to enable experimentation with new business models, taking into account requirements of low entry costs for suppliers and adequate provision for privacy."

The focus of CPA 12 is on:

- Development of open and reusable middleware for the efficient and flexible deployment of application services leased to an organisation and making use of external resources, possibly in combination with internal resources. The reuse of computing and data grid research prototypes and their integration with existing application development frameworks (CORBA, Java, etc.) could also be included.
- Development of service management frameworks with adequate provision for resilience, persistence, security, confidentiality and end user privacy.

The concrete objectives of the TAPAS project closely match this general objective of CPA 12 and its focus. Organisations are increasingly focusing on their core businesses and streamlining their operations by 'outsourcing' non-core businesses to external organisations. In particular, many organisations find it cost effective to outsource their IT applications to Application Service Providers (ASPs). As organisations become global and distributed, centrally managed hosting solutions currently provided by ASPs will need to be replaced by multi-site, distributed hosting solutions. The TAPAS partners believe that an ASP will increasingly be called upon to host distributed applications that make use of a wide variety of Internet services provided by different organisations. However, to work effectively, ASPs must guarantee security, provide resilience and service level agreements over commonly available infrastructures. Being a cross programme action, the objectives of TAPAS also match a number of *action lines*:

- *II.2.2 Smart Organisations*, with focus on "development and validation of novel architectures, software platforms and pre-standards to support interoperability, seamless integration and knowledge sharing between heterogeneous enterprise applications and services".

- *II.4.1 Trust in information infrastructures*, with focus on "development, integration and validation of trust and security technologies in information infrastructures"

- *IV.2.1 Real time distributed systems*, with the objectives "to develop and assess models, technologies and tools for sharing and interactive use in real time of applications and resources in geographically dispersed locations…", with focus on "development environments to support real time, distributed applications".

- *IV.3.1 Software architectures*, with focus on "models and notations for describing systems architectures and being able to reason about them. The main concern is to guarantee required quality attributes (for instance on scalability, performance and reliability) of systems."

- *IV.6.2 Interfaces and buffers for seamless end to end services*, with focus on "network programming interfaces".

Quoting again from the Council Decision on the Fifth Framework Programme:

"Realising the full potential of the information society requires technologies, infrastructures, applications and services, accessible and usable by anyone, anywhere, anytime, whether it be for business or individual use. Collaborative research and technological development is needed to create both the critical efforts and the interoperability necessary to ensure this in Europe"

The challenge of developing and validating "open architectures, technologies and tools to allow for the provision of a variety of applications as networked services" is so enormous that no single research institution is in a position to tackle this problem on its own. The TAPAS consortium brings together academic researchers from software engineering, middleware and Internet communities to work with an ASP. The inclusion of an ASP in the consortium will speed up the migration of TAPAS results and technologies to industry. At the same time, the ASP, being a small to medium scale enterprise, is unable to undertake research work on its own on the next generation of application hosting technologies, so will benefit from collaborative work. Success in this endeavour will substantially ease the development of domain specific application services; for example, services for *computational grid* (enabling scientists to move large amounts of data globally and perform long lasting parallel scientific computations), *electronic markets* (enabling art dealers to conduct real-time auctions on a world-wide scale) and *collaborative applications* (enabling globally scalable multi-person games), to name just a few.

An effective technology for application services provision is likely to reduce the significant manpower needed for in-house development, deployment and management of IT applications as it enables organisations to focus on their core businesses and outsource their applications to ASPs. It thus has a potential to increase the productivity in the booming service sector where qualified manpower is scarce, creating new business prospects with new employment opportunities.

7. Contribution to Community social objectives.

Concern for Community social objectives strongly motivates the aims of the IST Key Actions. In case of Key Action II "New Methods of Work and Electronic Commerce":

"to develop information society technologies to enable European workers and enterprises, in particular SMEs, to increase their competitiveness in the global marketplace, whilst at the same time improving the quality of the individual's working life, through the use of information society technologies to provide the flexibility to be free from many existing constraints on both working methods and organisation, including those imposed by distance and time. Specific attention will be paid to the social implications of new working methods, in particular their impact on equal opportunities and quality of life".

A Cross Programme Action project addressing this area, and with the objectives of TAPAS, has an important role in contributing to the Community's social objectives. Namely, it can bring improvements in several aspects including, but not limited to: the quality of life; employment; enforcement of ethical values, such as privacy. One of the objectives of the Community is the fulfilment of the "information society". The quality of life of the people is today intimately connected to their ability to access information, despite their geographical location, transportation facilities, etc. Furthermore, this view can be enriched by considering the ability to access both information and services, and to the ability to access them "anytime-anywhere".

Most employment created in Europe resides within small and medium sized enterprises (SMEs). SMEs may be threatened by the current trends towards company mergers and as such they should seek competitive advantages that relate to their small size: heterogeneity, proximity and geographical coverage; agility and versatility; lightweight structure. As examples, SMEs should: a) achieve critical mass for efficient market access, b) develop co-operative research and development facilities, c) reduce financial and administrative overheads, d) increase production flexibility, e) shorten development and marketing periods, f) contribute to product and process innovation, etc. Most or all of these desirable initiatives depend on the ability of SMEs to master IT technologies, and be connected. Awareness about the best use of IT in general, and creation of virtual enterprises or enterprise networks are two major key factors of success of SME-based enterprise tissues.

However, SMEs are increasingly finding it difficult to develop, maintain and manage their IT applications largely due to difficulties in retaining and attracting trained IT staff. Application Service Providers (ASPs) hold the promise of providing an attractive solution by making available application hosting facilities on remotely managed servers. However, to work effectively, ASPs must guarantee security, provide resilience and honour service level agreements over commonly available infrastructures. Furthermore, ASPs need to ensure that hosted applications are capable of accessing a wide variety of services irrespective of the platform or the organisation through which they are provided. The project will develop and validate open architectures, technologies and tools to allow for the provision of a variety of applications as networked services over a commonly available infrastructure. This project seeks results that help to realise this goal, by looking at ways to architect and program large-scale, Internet-based systems and services:

- that are reliable and secure, both for the provider and the users;
- yet made of COTS technologies wherever possible, so that they are also cheap;
- and finally with interfaces that hide the complexity of the functional and non-functional properties, so that they are easy to use.

One of the reasons that make this project challenging is that these objectives are very often seen as contradictory.

8. Economic development and S&T prospects

8.1 Introduction

It is clear that the Internet and associated information technologies are going to be central to the fabric of society in the 21st century. Organisations are increasingly using the Internet for their day to day functioning. This use includes interactions between customer-to-business organisation (e.g., product support, customer complaint handling) as well as between business organisation-to-business organisation. This naturally leads to interactions that cross organisational boundaries. An interesting development has been for organisations to focus increasingly on their core businesses and streamline their operations by 'outsourcing' non-core business to external organisations. In particular, many organisations find it cost effective to outsource their IT applications to Application Service Providers (ASPs). An ASP typically uses middleware and component technologies for deploying, hosting and managing applications of an organization from a centrally managed facility. However, as organisations become global and distributed, such centrally managed hosting solutions will need to be replaced by multi-site, distributed hosting solutions. ASPs are therefore under increasing pressure to be able to host application services that are capable of finding, purchasing and managing services performed by other organisations. The TAPAS partners believe that ASPs will need hosting platforms and solutions that are:

- Highly available, to meet the needs of today's global business environment.
- Secure, to protect the privacy of users and the integrity of the enterprise.
- Reliable and scalable, to insure that business transactions are accurately and promptly processed.

With the above observations in mind, the TAPAS project seeks to develop novel methods, tools, algorithms and protocols that support the construction and provisioning of Internet application services. TAPAS will be developing QoS enabled middleware services that will enable components to be deployed and interact across organisational boundaries. The results of this project will be to provide highly assured mechanisms, protocols and architectures that will help in the deployment of Internet services in Europe. Alongside this will be tools and techniques specifically tailored for the design and evaluation of Internet application services. In order to ensure that those benefits are realised, we need to foster their deployment. This can be achieved in two distinct ways, namely dissemination and exploitation. We describe these in detail in the following subsections.

8.2 Dissemination

By dissemination, we mean making the results of this project visible to a wide audience. This will be achieved by the following means: (i) publishing and presenting results within the scientific community, (ii) providing input to the open source software development movement, (iii) influencing the relevant standardisation bodies, and (iv) training of students. These points are discussed below:

(i) The research results will be presented at the leading international conferences and in the leading scientific journals to generate a level of awareness and constructive feedback from the scientific community and the industrial research community. The relevant scientific community is in fact at present really a set of almost entirely distinct communities, namely the fault tolerance, distributed computing, software engineering, middleware and computer security communities, each with their own journals, conferences, and workshops. We will therefore seek not only to disseminate our results in these separate communities, but also to contribute energetically to conferences and workshops that seek to attract members from all these communities. Such events include, IEEE/IFIP International Conference on Dependable Systems and Networks, IEEE/OMG

International Enterprise Distributed Object Computing Conference, IEEE Intl. Symp. on Object-oriented Real-time distributed Computing, IFIP Conference on Distributed Applications and Interoperable Systems and IFIP Intl. Conference on Distributed Systems Platforms and Open Distributed Processing. Members of TAPAS frequently serve as organisers and programme committee members of these events. Information dissemination will also be carried out through the events organised by the IST sponsored network of excellence on distributed computing, CaberNet, of which many TAPAS partners are members. In this respect, TAPAS will also cooperate with related IST projects, such as CPA2 projects on dependability: MAFTIA and DSOS; both of these projects are coordinated by Newcastle, so cooperation is guaranteed in this particular case.

(ii) The recently-initiated Robust Open Source movement provides a very interesting new means by which the results of our research could reach, and be taken advantage of by, a very wide audience. It aims to build on the success of the open source movement, on whose products much of the Internet already relies very heavily. The TAPAS project is in a good position to contribute as TAPAS middleware services and architectures will be developed using open source application servers and widely used component technologies such as CORBA and Java. Software deliverables of the project will be distributed in open source form.

(iii) The likely future effectiveness of the many and various official and ad hoc standardisation bodies that concern themselves with issues of relevance to our research is difficult to judge, and it would be naive for us to plan now how best we might seek their approval of the project's intended results. However, we have experience of contributing to standardisation bodies such as the OMG and IETF. For example Newcastle played an active role with IBM, IONA Technologies and others in making an OMG submission "Additional structuring mechanisms for the OTS"; this submission has now been adopted as the OMG standard.

(iv) Master and PhD students that work in the project and have an intimate know-how in the technology will be encouraged to take up positions in industry to act as "evangelists" after they finish their university training.

8.3 Exploitation

In addition to the general dissemination activities outlined above, we have many exploitation routes for TAPAS results through our on going research projects and interactions with the advisory panel. We first describe how the academic partners will exploit TAPAS results and then describe the benefits gained by the industrial partner.

Newcastle University: The Distributed Systems Group has a strong record of working with industries. In conjunction with Nortel (Harlow research lab), we contributed to the development of the workflow standard by making a submission to the OMG based on our workflow technology (Nortel and University of Newcastle upon Tyne, "Workflow Management Facility Specification", Revised submission, OMG document bom/98-03-01). We worked with IBM (Winchester lab) and IONA Technologies towards the development of new transaction standard (contribution to OMG RFP, "Additional structuring mechanisms for the OTS"). Our technologies have been in use in several industrial settings. Results from TAPAS will be used in existing and future research projects on middleware related distributed computing.

Arjuna Solutions Ltd (ASL) was formed on October 1998 in order to exploit the products, technology and expertise created by the Group. It is based in Newcastle, and began its life as privately owned, primarily by the five founding members, who are all researchers in the Group. On July 2000, ASL became a wholly owned subsidiary of Bluestone Software Inc. In October 2000, Hewlett-Packard Company (HP) and Bluestone announced that the companies have reached a definitive agreement under which HP will acquire Bluestone in a stock-for-stock strategic transaction. Bluestone's Application server and Arjuna's transaction and workflow technologies will form parts of HP's middleware services for e-commerce.

HP-Arjuna Lab is now a major employer of skilled manpower in the North East of England. Of the five founding members of Arjuna Solutions Ltd., S.J. Caughey is now Director of the HP-Arjuna Lab, M. Little, S.M. Wheater and D. Ingham are HP Distinguished Engineers at the Lab and Santosh Shrivastava is back at the University, leading the distributed systems research group. Close industry-university collaboration is guaranteed, and a joint research project on B2B e-commerce has been initiated. HP-Arjuna will be represented on the Industry Advisory Board.

Bologna University: The research group at the Department of Computer Science of the University of Bologna maintains close cooperation with national and international industries, including Microsoft (Cambridge Reasearch Laboratory) and Sun Microsystems, and national research institutes, the "Fondazione Marconi" and the ENEA (the Italian bureau for new technologies applied to the energy and environment), in the form of joint investigations under contracts and grant programmes.

In addition, this research group is going to be involved in the national project entitled "Infrastructure Support for e-business applications", that will be carried out in collaboration with a number of Italian Universities and companies. This project, which is currently under formal approval by the Italian Ministry of the University and Scientific and Technological Research, will greatly benefit from the results of the TAPAS project. In addition, two further projects, in which the research group in Bologna will be involved, can benefit from the TAPAS project results. Both these projects, entitled "A Distributed Broker for Quality of Service", and "Middleware for advanced services over large-scale, wired-wireless distributed systems", respectively, will deal with issues of QoS at the middleware level. These two projects have been recently submitted for approval to the Italian Ministry of the University and Scientific and Technological Research; both these projects will be carried out in collaboration with other Italian Universities and companies.

The cooperation and collaboration channels mentioned above will be to transfer the results that will emerge from our research activity in the TAPAS project.

University College London: UCL relies to a considerable extent on direct industrial funding and consulting. The knowledge required to be able to provide high-quality consulting services is often produced in projects such as TAPAS. UCL therefore hopes to exploit the results of the TAPAS in the following ways:

- Technology transfer initiatives towards the industry.
- Provision of consulting services to external companies (including education and training).
- Dissemination (consisting essentially of publications and courses).

The Software Systems Engineering Group of UCL is well positioned for this exploitation. It has ongoing research collaborations with a number of industrial partners, including British Telecom, Hewlett Packard Labs, IBM Hursley Park, Kodak, Microsoft Research, UK National Air Traffic Services, Philips, Searchspace, Telelogic, Toshiba Corporation, UBS Warburg, Unipower Solutions and the Zuehlke Technology Group. The TAPAS project partners hope to use these good collaborations as a route for exploiting the knowledge produced in TAPAS.

Cambridge University: Cambridge University has a plethora of industrial collaborators, and will seek to exploit any and all of its research when appropriate. In this project, links with Microsoft Research and Marconi Research may prove extremely valuable, as both have research laboratories in Cambridge working in collaboration with the University. As well as this, the Compaq systems research lab, and Hewlett Packard may also be potential paths to exploitation given long term relationships with both. Finally, we will be able to use the output of TAPAS within the academic community itself directly to support more performant application services for teaching and research.

Adesso AG: Adesso AG is a full service provider for the design, development and operation of e-business applications. The development paradigm applied is that of component-based software development. This paradigm and the application domain of e-business applications perfectly match because most e-business application encompass various COTS components. This does not only pose some extra challenges with respect to system integration, release management and test of e-business applications, but it is also hindering the business model ASP for e-business applications. Due to the heterogeneity of e-business application standard ASP service level agreements usually cannot be applied. Instead it is necessary to relate service level agreements to components of an e-business application individually. This may, for example, mean to define service levels agreements as the following:

- The portal site will be accessible for 98% of the time.
- Access to the e-controlling component is ensured for 90% of the time.
- The minimal recovery time for the access to individual customer data is 20 minutes, the recovery time for profile data is 60 minutes.

This example shows, that different types of functionality ask for detailed agreements. Thus, fine-grained service level agreements help to provide the services needed at affordable costs. Of course, it is possible to offer only more coarse-grained service levels, but this usually leads to cost explosions which are not acceptable for customers. With the possibility to define and implement fine-grained service level agreements developed by TAPAS, Adesso can foster its core business in several ways:

1. It is possible to argue for component-based development of e-business applications, because this is a prerequisite for fine grained service level agreements.

2. The range of software systems which can be integrated into e-business applications which are to ASP-operated is extended. For the time being, systems whose low robustness endanger the availability of the overall e-business application cannot be integrated. If it was possible to agree for lower services or such a component, it would be possible to integrate despite its robustness.

3. The ASP services of Adesso will be much more attractive, if fine-grained agreements are possible. In contrast to standard offerings, the ASP levels can be precisely adapted to customer requirements.

While the first and second way to foster the Adesso business cannot be calculated in concrete numbers, the third way is supposed to allow an extra 20% growth in ASP business (after being able to define and implement fine-grained service level agreements).

9. Workplan

9.1 General Description

The work of TAPAS will be structured into four technical workpackages in order to achieve the three operational goals stated in Section 2:

- WP1 (Application Service Requirements and Specification) will meet the goals related to SLA specification, Service Composition and Analysis Techniques;
- WP2 (Design of QoS-aware Infrastructure for Application Hosting) and WP3 (Implementation of QoS-aware Core Services) together will meet the goals related to Trusted and QoS-aware Services for Application Hosting; and
- WP4 (Assessment and Evaluation) will meet the goals related to assessment.

The fifth workpackage deals specifically with project dissemination and exploitation and the sixth workpackage deals with project management and coordination.

Basically, within WP1 we will work towards acquiring an understanding of the requirements and then develop SLA specification and its QoS analysis tools and techniques; WP2 will be devoted to the development of trusted and QoS-aware middleware architecture based on the requirements generated from WP1. WP3 will implement the architecture developed in WP2, and WP4 will perform evaluation and assessment of the architecture and its implementation primarily through demonstrator application building exercises. Each technical workpackage has a workpackage leader (lead contractor) responsible for overall direction and production of the deliverables. Furthermore, each workpackage has been divided into a number of related tasks, with a partner with specialist expertise acting as the task leader.

TAPAS will make use of open source UML tools (such as Argo/UML tool) for the development of SLA specification and QoS analysis tool and open source application servers (such as Jboss [7] or Jonas [8]) for the implementation of TAPAS architecture. Software deliverables of the project arising from this work (in particular deliverables D4, D8, D9 and D10) will be made available as open source software.

An innovative aspect of this project is its emphasis on hosting of important class of large scale networked *multi-party applications* that are characterised by a group of entities requiring many-to-many interactions; examples include collaborative applications (e.g., multimedia conferencing, multi-person interactive games), electronic share dealing and auctions, and 'information supply chain' applications typically based on publish-subscribe paradigm. Currently, strict separation exists between the middleware that executes application services and the network, so providing services that go beyond 'the best effort' is difficult. The next generation Internet is expected to offer services to users to enable them to request and reserve network resources that their applications require. However, application developers will have to deal with the complexities of interacting with communication services for QoS allocation. It is not entirely clear how this interaction should take place, and research work is required on specification and enforcement of QoS at the network level. TAPAS will examine ways of 'opening the network cloud' and develop network QoS API for multi-party traffic control. For this reason, TAPAS workpackages contain specific tasks that relate to network level QoS issues.

9.2 Workpackage list

Work-package No	Workpackage title	Lead contractor No	Person-months	Start month	End month	Phase	Deliv-erable No
WP 1	Application Service Requirements and Specification	4	68	1	30	R	D1, D2, D3, D4
WP 2	Design of QoS-aware Infrastructure for Application Hosting	1	68	7	3 6	R	D5, D6
WP 3	Implementation of QoS-aware Core Services	3	95	7	36	R	D7, D8, D9, D10, D11
WP 4	Evaluation and Assessment	2	60	7	36	R	D12, D13, D14, D15
WP 5	Dissemination and Implementation	1	10	1	36	R	D16, D17, D18
WP 6	Project Management	1	58	1	36	R	D19, D20, PP1-3, PM1-3
	TOTAL		359				

This section describes the various workpackages listed in the table. For each workpackage, we highlight the main objectives, the novel aspects of the tasks to be carried out in it, and the expected results. Section 9.3 summarises this discussion by highlighting in a condensed way the main objectives, deliverables and milestones of each workpackage.

9.2.1 WP1: Application Service Requirements and Specification

This workpackage, led by UCL, has been divided into four tasks. The first two tasks study, respectively application hosting requirements and networking requirements for many-to-many communications. A common deliverable report (D1) will document the results from the study. These results will feed into WP2 and WP3. The remaining two tasks concern SLA specification and analysis: alongside functional components interfaces, these SLAs specify the non-functional qualities of service, such as the availability, performance and scalability characteristics of components; analysis tools will be developed for reasoning about the performance and scalability characteristics of components and their composition. Deliverable report D2 will document the SLA specification method, report D3 will describe service composition and analysis method developed within the project and D4 will be the corresponding toolkit. The toolkit can be used to confirm that the various SLA commitments can be honoured for a variety of physical configurations and load conditions *before* commencing the actual deployment of the application. The results from D1-D4 will be used for the development of applications in WP4.

Task 1.1 Application Hosting Requirements

Industrial partner Adesso will lead this task. The basis for defining requirements will be the analysis of the needs of the different parties in application hosting, and limitations of current application hosting from the views of the different parties. Such limitations typically arise from core problems of the ASP approach such as the risk for a client to loose control over an application [1]. In order to keep the requirements discussion close to industrial needs we will develop scenarios for QoS aware applications. Such applications shall not only overcome existing problems, but will provide new features, which might in the end lead to new unique selling propositions of QoS-aware ASP and thereby to an increasing ASP market.

Electronic marketplaces introduce a huge number of different aspects regarding QoS. While customers usually wish to have control on their private data, the providers are interested in secure transactions together with data protection, as competitors should not be able see offers. Furthermore the company running the marketplace as a business might be interested in integrating many different providers, secure transactions and as well in anonymity of customer and provider during the ordering process in order not to be bypassed. In a QoS aware application scenario such a company could even supervise the ASPs fulfilment of the SLAs by technical means. However, the ASP will be interested in using services like storage area networks from other service providers, to monitor the behaviour of the applications etc. A condensed scenario containing the marketplace's QoS issues are online auctions, because the same participants as in the general marketplace scenario are involved and even more requirements like delivery time constraints for offers and bids will come up. Especially on a B2B background a QoS-aware application for auctions may even become a part of a real client's marketplace, if it proves to be valuable.

SLAs define probably in most cases a relationship between a service client and a service provider. However, some clients like insurance companies or small financial institutes may not be not able to supervise the fulfilment of the SLAs due to lack of technical knowledge and staff. This can lead to severe acceptance problems, especially in the area of trust delegation and management. An insurance company could for instance state in a SLA that data has to be stored encrypted in a storage area network or data centre. Without provider-independent means to supervise this feature the company would most likely not choose the ASP solution due to an obvious and understandable lack of trust. In order to solve problems of trust management and fulfilment of SLAs the discussion of the scenarios might identify the need for additional participants, who will mediate between service user and provider, hence having a kind of referee or notary role. While the main goals of the TAPAS project aim on technology-related improvements the introduction of a new role into the ASP world has to be investigated carefully, because the applicability of the scenario could be based on a role, which must be brought to the markets.

Task 1.2 QoS Networking Requirements

Partner UCAM-CLAB, who is a member of the Internet Architecture Board and with expertise on reliable multicast protocols for the Internet will lead this task. This task will study network QoS requirements for supporting multi-party applications that are characterised by a group of entities requiring many-to-many interactions. The 'best effort' service model of the Internet makes no guarantee about the QoS. The Internet community has made two efforts to develop standards for Internet Protocol (IP) with QoS capabilities: the integrated services architecture (IntServ) and the differentiated services architecture (DiffServ). In the absence of router implementations as required by IntServ, the community has moved on to DiffServ, where there is only a per-class scheduling problem. However, in this arena, people have struggled, even with complex "edge" traffic conditioning, to find a scheduler and allocation and service specification that can achieve a delay bound.

The network provider usually carries out long term provisioning of their network (Sprint, Worldcom and AT&T claim they upgrade their Internet capacity by 100% every 6-9 months). This is based on measurement of service use and service violations. Due to the rapid evolution of applications in the Internet, short term provisioning is also carried out - this is known as Traffic Engineering, and often takes the form of re-balancing routes, and of re-prioritising different less important applications (e.g. downgrading mp3 music download traffic for example).

Multi-party applications (trading, games, online debating systems), with QoS requirements represent another stage in the evolution of Internet traffic patterns, in that we would expect a high degree of correlation in the traffic from different sources that make up the applications. To this end, the resource allocation that is normally made on the timescales of traffic engineering now has to be carried out on the timescales of application sessions. Of course it is possible that these applications will be long-lived (although evidence from game data is against this).

This part of the research will study the traffic patterns from these applications, so that the information could be applied for developing algorithms and signalling API for traffic engineering that can operate at the appropriate time scales; this will be addressed in task 3.2

Task 1.3 Specification of Service Level Agreement

Partner UCL with expertise in system specification, analysis and modelling will lead this task. Rather than defining yet another modelling language for application services, the TAPAS partners intend to adopt the UML as the baseline for the description, modelling and analysis of application services and extend the UML with formally defined stereotypes and properties that support the modelling of distributed interaction primitives and QoS characteristics. This approach is motivated by the large number of users of the UML in industry, who will find it easier to adopt a small UML extension than an entirely new modelling approach. Moreover, there are various very good UML tools available as open source products and we plan to use the Argo/UML tool as a baseline for the tool development of TAPAS.

The different diagrams provided by the UML are sufficient to model the functionality of application services, but the lack of a formal UML semantics inhibits reasoning about qualitative properties of the design of distributed application services, such as absence of livelocks, deadlocks or safety properties. Moreover, the UML does not support the modelling of QoS characteristics, such as the performance, scalability and availability of distributed application services. In [22], we have identified that there is only a limited set of interaction and threading primitives available for distributed objects and we have expressed them as stereotypes of UML models. In particular, we have defined synchronization stereotypes to be used in state diagrams for synchronous, oneway, deferred synchronous and asynchronous communication that is available in distributed object middleware (e.g. CORBA, COM and Java/RMI) and class diagram stereotypes for the different threading policies supported by middleware object adapters (such as CORBA's POA). We propose to use the same idea to the component-based middleware standard that TAPAS will adopt and define suitable stereotypes and properties.

Task 1.4 Service Description, Composition and Analysis Techniques

Partner UCL will lead this task. The semantics and properties of the stereotypes will be defined by mapping stereotyped UML diagrams to process algebra, such as CSP [23], CCS [24] and FSP [25]. The choice of a process algebra is motivated by its algebraic properties, most notably composition, its precise operational semantics based on labelled transition systems and the availability of powerful model checkers, such as the FDR tool for CSP [26], and the LTSA for FSP [25]. These model checkers apply compositional reachability analysis in order to analyse qualitative properties of application service provision, such as adherence to safety properties and absence of deadlocks and livelocks. In [27], we have used this technique successfully to model check labelled transition systems that were derived from stereotyped UML class and state diagrams for the design of distributed objects against absence of deadlocks. In TAPAS we plan to extend this technique of generating an equivalent process algebra specification for a UML design to model and verify general liveness and safety properties of the design of distributed application service components.

In addition to the analysis of the qualitative properties discussed thus far, we would also like to be able to reason about the quantitative properties of application service design, such as performance, scalability and dependability prior to building a new or modifying an existing service. For these quantitative properties we will take a similar approach of expressing them as annotations to UML diagrams and then derive a formal representation from these annotated design diagrams that is amenable to quantitative analysis. We propose to use Markovian stochastic process algebra, such as MPA [2], PEPA [4] or TIPP [5]. The choice of stochastic process algebra is motivated by the fact that they are direct extensions of the approach that we used for qualitative properties and possess the same algebraic property of compositionality, they can formally express the timed and stochastic behaviour that we would like to model and finally there are analysis tools available, such as the PEPA Workbench [3] and the TIPP tool [6] that use continuous timed Markov chains as the underlying formalism for reasoning about utilization, response time, scalability and dependability.

9.2.2 WP2: Design of QoS-aware Infrastructure for Application Hosting

TAPAS architecture will use SLAs not only as an inter-organizational contractual feature but also to govern component execution. To this end, QoS negotiation, establishment, and adaptation facilities will be added to the TAPAS middleware and these will be used by component containers to make them QoS aware. This workpackage, led by partner UNEW has four tasks: middleware architecture, trust management, network control architecture and component execution environment. These will be initiated after the first six months of the project, by which time results from WP1 (requirements deliverable report, D1) will be available to guide us. These tasks will last for six months each; the result will be the deliverable report D5, available by the end of the first year of the project, that will describe the interim TAPAS architecture for application hosting. Work on D5 will influence the implementation work of WP3 that will run concurrent to WP2. We expect that case studies and evaluation work (WP4) will provide us with valuable feedback on the effectiveness of the TAPAS architecture of D5, suggesting improvements and modifications. For this reason, the middleware architecture task will be resumed during the last six months of the project, and a revised architecture report will be produced (deliverable report D6).

Task 2.1 Middleware Architecture

Partner UNEW with expertise on dependable middleware services and architectures will lead this task. As we stated in section 2, component oriented technologies (such as CORBA components, EJBs) support the cost effective creation of services by composing and integrating existing components [28, 29]. Current component technologies support the specification of functional component interfaces. They, however, do not adequately support the definition of the non-functional characteristics -QoS- of component execution [30, 31]. How to incorporate performance oriented requirements (e.g. timeliness of execution and relative processing speed requirements), reliability oriented requirements (e.g. high availability and relative failure recovery requirements), and security oriented requirements (e.g. authentication, privacy) in the middleware?

Component oriented middleware promotes the use of *containers* to host component instances [32-34]. A container is a runtime environment for component instances and their homes. Several containers could be hosted by a same *component server*. A container is more than a simple execution environment. Containers hide the complexity of most of the system services like transaction, security, persistence, and notification services. Thus, containers take part in the management of non-functional aspects of a component. To this end, QoS negotiation, establishment, and adaptation facilities will be added to the underlying services (see WP3) and these will be used by component containers to make them QoS aware. That is to say, a container will have a *QoS contract* with the underlying service, that the component execution environment will (try to) honour. Specification of these contracts will be derived from the application level SLAs using the techniques developed in WP1. This task will examine overall architectural issues in the development of such containers and their execution environment. Given the emphasis of the project, particular attention will be placed on issues of trust and how it impacts inter-organisation interactions and multi-party communications.

Task 2.2 Trust Management

Partner UNEW will lead this task. We are assuming that the organisations involved might not trust each other, so an important part of our work will be concerned with developing infrastructure support for interactions between two or more mutually distrusting organisations. A fundamental approach is to use mutually trusted third party (TTP) to create a context in which multi-party interactions can be carried out. An interaction is assumed to involve the electronic exchange of items of value (information, goods, services). Governed by agreed rules, these exchanges involve coordinated changes to the state of shared data and the imposition of related obligations (and restraints)on the parties to the exchange. Few assumptions should be made about the nature of the interaction or about the relationship between parties. For example:

- an interaction may be short- or long-lived: from fulfilment of an order to provision of a complex, evolving service;
- interactions may be composed from sub-interactions: service delivery may involve the sub-contracting of obligations to other parties whilst ensuring compliance with the obligations imposed by the "parent" interaction;
- interactions may be dynamic: the rules governing the interaction may be subject to re-negotiation and the parties to the interaction may change;
- the relationship between parties may be asymmetric (customer-supplier) or symmetric (peer-to-peer)
- parties may be strangers to each other or there may be a pre-existing basis for trust. Trust between parties may develop as a result of the interaction. Trust relationships may be symmetric - mutually trusting or distrusting - or asymmetric, when trust (or distrust) is not reciprocated.
- an interaction may be subject to both global constraints and to local constraints with a requirement to resolve conflicts between the two.

Parties engaged in an interaction require an infrastructure that both facilitates the interaction (for example, to share and coordinate application state) and provides guarantees with respect to protection of their individual and collective interests. The infrastructure should support interaction in a hostile environment: interaction, between strangers over open networks. These are very demanding requirements and the subject of trust management is a relatively new area of research [10-14, 36]. In this task we will investigate various design options, such as those discussed in [15-19,35] for developing the basic building block of non-repudiateable update of and access to shared objects; we have done some preliminary work in this area [16].

Task 2.3 Network Control Architecture

Partner UCAM-CLAB will lead this task. From an application's perspective, the network is just another type of resource that needs to be managed. The goal here is to work bottom up from the network layer looking at fault tolerance and performance management mechanisms, up to the level of service interface and behavioural description that matches the work done in the other work packages; the intention is to make the network services for multi-party communication all available at the object level as first class objects in the system, so that overall distributed services can be built out of components as needed, and with no special different treatment of networking entities from other forms of distributed object engineering.

This task will build on the work of UCAM-CLAB [37, 38] and will look at existing control and signalling protocols including recent work on RSVP/IntServ/Cops subscribe and broker protocols for DiffServ and others in the IETF siglite work. As stated earlier, results from task 9.2.3 will be applied for developing algorithms and signalling API for traffic engineering, collectively referred to here as the network control architecture. The architecture will include feedback mechanisms for notifying applications when performance is below the SLA level, or when an application violates the SLA.

Task 2.4 Component Execution Environment

Partner UNiBo with expertise on QoS middleware will lead this task. As indicated earlier, component containers will have QoS contracts with the underlying services, that the component execution environment will (try to) honour. Our aim is to ensure that a developer only has to write the functional code in order to complete the component implementation, all the non-functional properties will be derived from the SLAs and used to configure the execution environment at the time of component deployment. This will require operations to discover component ports in a generic manner (same operations for any component type), or in a specific one (operations generated according to the component type); these operations will be used by the execution environment to introspect and interconnect component instances at runtime [34]. Thus our environment will require reflective capabilities [39,40].

In existing component oriented middleware systems, Component Descriptors are used for recording information about the component implementation. We will examine what additions are required for extending these descriptors with QoS and trust relationship policy descriptions (borrowing ideas from [41]) to enable run time generation of configuration information.

9.2.3 WP3: Implementation of QoS-aware Core Services

Partner UNiBo will lead this workpackage. This workpackage will implement a collection of QoS enabled services as required by the architecture developed by WP2. The overall implementation framework will be defined by the first task, QoS-aware containers, that will run concurrently with WP2 and will run initially for six months. The deliverable report, D7 will define the APIs. Three main implementation tasks, on group communication, trusted coordination and QoS monitoring will last for 18 months and will implement containers enriched with a particular functionality (group communication, trusted coordination and monitoring respectively). The deliverables, available by the end of the second year of the project will be working prototypes, deliverables D8, D9 and D10. The first task will be resumed during the last six months of the project and will produce the revised version of D7 (report D11), that will take into account the prototype design and implementation work. TAPAS will make use of open source application servers (such as Jboss [7] or Jonas [8]) for this workpackage.

Task 3.1 QoS-aware Containers

Partner UNiBo will lead this task. Applications require the ability to request resources based on their immediate usage needs. Fulfilment of a given resource request will usually require several interactions with the underlying service, and each service may well have its own specific *signalling methods and protocols* (for end to end QoS negotiation) for resource reservation and feedback and adaptation, using advance exception handling techniques [42].

Adaptation based approach requires implementing mechanisms that periodically monitor the resources availability of the execution environment, and provide the applications with appropriate adaptation and reconfiguration properties. Relevant instances of the adaptation based approach are the QuO [43], TAO [44], and DaCapo++ [45] and Agilos [46] projects. In these projects, the adaptation middleware reconfigures itself to provide the application with a transparent and stable execution environment. In summary, it is worth pointing out that adaptation based middleware architectures demand the following three principal stages of run time support: (i) *probing* the performance of QoS parameters, (ii) *instantiating* the initial middleware configuration, and (iii) *adapting* to on-the-fly variations.

This task will develop APIs for QoS-aware group communication. A different style of API is required for interacting with the trusted coordination service, where negotiations and instantiation will be for establishing appropriate bindings with TTPs , as discussed next.

Task 3.2 QoS-enabled Group Communication

This task will be led jointly by partners UNEW and UCAM-CLAB. Group communication protocols play an important part in the construction of services that need to be replicated for fault tolerance and/or load balancing reasons [47-50]. Group communication protocols enhance multicast by providing guaranteed message delivery, atomicity and total ordering (ensuring all intended recipients receive the same messages in the same order); existing systems do not support QoS enabled communication (properties of timely delivery to all participants, large group size) over wide-area networks. Another class of group communication protocols use publish/subscribe model for handling event based communication as required in say share dealing. Existing publish/subscribe middleware services, such as CORBA event service [51], Java Message Service, JMS, [52] do not adequately support the notion of QoS, neither do the research systems developed so far [53,54]. This is because, these systems assume a 'best effort' transport layer. In this task we will develop a QoS enabled transport protocol for the Internet and use it to support group communication systems for the Internet such as the one developed by partner UCAM-CLAB [55-57]. The work will be extended to support a suitable JMS implementation. We will examine work being done in this area by IST projects such as CORTEX.

Turning our attention to the development of a QoS enabled Internet transport service itself, we note that the protocol PGM [58] implements explicit "due date delivery" semantics. We thus have the beginnings of one-to-many protocol support for homogeneous and heterogeneous rate controlled, reliable delivery of data, with control over the timelines with which a specific item of data is delivered. Coupled with feedforward information signalled to a QoS sensitive network, and feedback from the network, a framework can be constructed that provides efficient communication; preliminary work in this area has been performed by partner UCAM-CLAB [37,38]. An important question for the network API designer is "how to encode efficiently a multiparty traffic specification" - clearly specifying the full matrix of source models is not going to be viable, but assuming all sources are homogeneous is also unlikely to meet the requirements found in WP1. The other side of this coin is that a multi-party application can have a partial QoS contract failure. This needs to be signalled efficiently to the relevant parties in the application by the network, and again we will investigate appropriate mechanisms.

Task 3.3 Trusted Coordination

This task will be led by partner UNEW. The infrastructure for trusted coordination should support the life-cycle of an interaction, which may include some or all of the following steps (repeated as necessary and, possibly, recursively during an interaction):

- *introduction*: the exchange of credentials to establish sufficient trust to pursue an interaction. This could range from the exchange of public keys to assertions of credit-worthiness, organisational membership etc.

- *negotiation*: reaching binding agreement on the rules that will govern the interaction (the contract) and deriving sufficient information from the agreement to be able to instantiate an infrastructure to support the interaction

- *instantiation*: the expression of the contract governing the interaction as a configured and instantiated infrastructure

- *activation*: execution of the activity governed by the negotiated agreement and facilitated by the instantiated infrastructure

- *assimilation*: the processing of interaction history to enable decision support to future interactions. For example, "reputation tokens" might be generated so that they can be used to establish trust in subsequent introduction phases.

We will begin by implementing two party client-server interaction with non-repudiation and authentication, based on the work reported in [16] and extend it to include TTPs. This framework will be further extended to support fair exchange protocols and multi-party communications. Interactions with the IST project MAFTIA will be relevant here.

Task 3.4 QoS Monitoring

Once a design has been achieved and been proven to have the desirable qualitative and quantitative properties, it can inform not only the implementation in terms of an appropriate component framework, but also its deployment. In particular, the performance parameters that have been modelled and been established to meet the quantitative performance and scalability requirements will then be used to inform the definition of SLA that govern the execution of components across organizational boundaries. The TAPAS component execution middleware API developed in task 3.1 will be used to monitor conformance of component execution with these SLAs.

It is our intention to use the data gathered during SLA monitoring to inform the refinement of model parameters. In particular, it will be useful to validate the assumptions that have been made during the initial design about the performance of operation executions and the load characteristics against data gathered during the performance. Already to date, application services are designed in an incremental and iterative manner. The SLA monitoring data gathered after the first version has been deployed will provide very valuable data for the tuning of quantitative design parameters in the UML model during future iterations.

9.2.4 WP4: Evaluation and Assessment

This workpackage will be led by the industrial partner Adesso. The TAPAS project will evaluate the methods, tools, algorithms and protocols that support the construction and provisioning of Internet application services that it will develop by comparing them to current state-of-the-practice (e.g. an off-the-shelf CORBA or J2EE application server). Our principle way of evaluation will be to develop demonstrator applications using the facilities of the ASP partner. The project will port the components of these applications to the TAPAS platform and compare their execution characteristics to those developed with off-the-shelf products.

We intend to build a particularly demanding application (in terms of QoS) that ASPs are currently unable to host using present day middleware platforms. The application we have chosen is that of electronic auctions. TAPAS partners UNEW and UNiBO have worked jointly on large scale, dependable auctions on the Internet [59,60], so this provides a good starting point. We will be able to demonstrate features such as the use of QoS enabled group communication for supporting timely delivery of notifications from the auctioneer to bidders, and supporting replication of auction servers for availability and so forth. These aspects are covered in tasks 4.1. The second task (task 4.2) will use the experience of building the auction application as a basis for evaluating the results of the TAPAS project. Deliverables will be three yearly TAPAS evaluation and assessment reports, D12, D13 and D14 and the prototype auction implementation D15. The first year report, D12, will take into account the results of the requirements report (D1) for developing the evaluation plan (setup, scenarios, metrics, measures), to be followed for second and third year work. All the reports (D12, D13 and D14) will include feedback from the Industrial Advisory Board (see below).

Task 4.1 Internet Auctions

The solutions developed by the TAPAS project will overcome limitations in current application hosting, not only for application service providers but as well for software developers and of course for the users of such applications. In a B2B marketplace scenario an auction application will provide a broad variety of quality and trust related aspects, because many different groups of participants are involved:

- Customers, who place bids for offered goods or services
- vendors placing offers for goods or services
- a company running the marketplace, i.e. the ASP client
- an application service provider hosting the applications
- service providers who offer technical or business services like data centres, credit card billing, logistic services etc.

All these different groups have different requirements regarding the quality of the application's services. The variety of requirements will be available from the requirements analysis performed in WP1 and will cover some basic topics such as (i) Trust management, (ii) Security policies, (iii) Privacy, and (iv) Measurable quality of the underlying network such as bandwidth, response time etc.

A vendor might, for instance, be interested in staying anonymous during the auction until the deal is settled, while the company running the marketplace will be interested in features such as availability and performance of the application. Currently SLAs are specified as contracts between two parties, i.e. in a non technical way. In order to increase the reliability of the SLA fulfilment and hence the acceptance of outsourcing even core parts of enterprise-relevant applications, SLAs clearly have to be specified on a technical level. The actual construction of the SLAs is part of the case study and will make use of the specification techniques and tools developed in WP1. We will identify use-cases from the perspectives of the different participants. Some coarse-grain use-cases are already deductible from the description of the participants above:

- a customer places a bid for an offer during an auction
- a vendor adds an offer for a good or service before the start of the auction
- an authorised ASP client staff member starts an auction
- the application service provider reconfigures the application to use a different service provider, for instance a cheaper data centre
- a service provider changes features of the service, which affect it's quality

These use-cases have to be refined in order to cover different behaviour of the system appropriately. The first result of the case study is an application, deployed and hosted on dedicated servers, which will be usable for a group of testers. The installation should sufficiently demonstrate the results from the perspectives of the different participants such as end-users, service providers etc. Furthermore, as a base for the following evaluation, the development artefacts such as use-cases and design models are available. Last but not the least, a summary of the development experiences will provide valuable input to the subsequent task of compiling an evaluation study and refinement of the TAPAS architecture, WP2.

Task 4.2 Evaluation

The goal of the evaluation study is to inspect the TAPAS methods, techniques and components from an industry point of view. The evaluation will utilise the results of the case study as a starting point. Part of the results is the application itself, i.e. the software that can be deployed and hosted. While the case study focuses on the actual realisation of an QoS-aware scenario, the evaluation has to investigate as well quality aspects of the rest of the life cycle. For instance the SLA modelling techniques and tools have to be considered as well. Again it is reasonable to start with the experiences from the case study, for which these techniques and tools will have been used during the construction. However, the case study cannot cover all aspects of a broadly usable technique. Therefore further effort has to be taken to explore uncovered aspects. This must start with identifying the methods, tools, techniques and components developed by TAPAS. We will scan the use-cases of the case study for test cases that involve QoS features and can then construct test cases.

In an auction scenario, which is embedded into a marketplace scenario, the set of QoS-related test cases might base on some of these cases:

- Offers of providers have to arrive at all the browsers of the customers, who take part in the auction, at the same time or at least within a predefined interval, depending on the specifications of the SLA. Here the SLAs for all participating customers must be equal regarding the interval. On the other hand all customers should have the possibility to place their bids until a specified point in time, again this may be specified as an interval. The interval is not necessarily restricted to small amounts of time like seconds. Especially in case of more complex goods like machines or insurance contracts the buyer might need a longer time for an analysis of the offer.

- A vendor wants to place an offer only during a certain period of time, e.g. each day during lunchtime in order to attract working people. Furthermore the vendor wants to stay anonymous for other vendors and even for the customers, at least until the deal is fixed.

- In a different scenario a vendor requires to include another service like credit rating in the whole business transaction to ensure that the buyer will be able to pay the invoice. Again the vendor wants to stay anonymous for the credit rating agency.

- The company running the marketplace, i.e. the ASP client, wants to ensure that all data is stored encrypted, if external data centres are used, but at least for credit card or payment data. Furthermore a maximum value for a system response time is specified.

- The application service provider is interested in a certain bandwidth for the connections to storage area networks or as well to the vendors and customers. Here the bandwidth is a means of implementation for the response time requirement.

- service providers who offer technical or business services like data centres, credit card billing, logistic services etc. may in contrast be interested in limiting a bandwidth or amounts of data transfers, because they serve multiple customers and face limited capacities of network and storage space.

The test cases then have to inspect the behaviour of the application and the middleware and network components. Therefore it is necessary to test the fulfilment of the requirements under normal conditions as well as under stress situations. Due to the variety of requirements such as guaranteed bandwidth, simultaneous arrivals, transaction safety including different systems etc. constructing stress situations will lead to a variety of possibilities: (i) limiting the network bandwidth, (ii) causing significant delays of messages, (iii) trying to access protected data, (iv) stopping features like encryption in underlying services, (v) accessing services beyond the asserted limitations of data amount or accesses. Thereby constructed test cases cover the fulfilment of the requirements. Furthermore the behaviour of the QoS monitoring components must be inspected, because the escalation of QoS failures is important to build confidence between the users of SLAs. Beside the test cases derived from the use-cases, some effort will be taken to attack the core features of the TAPAS components such as trust, security, privacy, network multicast etc. to find weaknesses and strengths, that have not been highlighted by the test cases.

Risk Analysis and Measuring Progress

The TAPAS aim of developing QoS enabled middleware services capable of meeting SLAs and supporting components that can be deployed and interact across organisational boundaries poses significant technical challenges. As previous sections have indicated, the project will have to innovate in the areas of architecture, system design and validation. There is inevitably a degree of uncertainty about how things will progress. For example, here are two areas of uncertainty identified earlier:

Trust management: Parties engaged in an interaction require an infrastructure that both facilitates the interaction (for example, to share and coordinate application state) and provides guarantees with respect to protection of their individual and collective interests. The infrastructure should support interaction in a hostile environment: interaction, between strangers over open networks. These are very demanding requirements and the subject of trust management is a relatively new area of research.

QoS-aware multiparty interaction: An important question for the network API designer is "how to encode efficiently a multiparty traffic specification" - clearly specifying the full matrix of source models is not going to be viable, but assuming all sources are homogeneous is also unlikely to meet the requirements identified in WP1.

The deliverables of the project by means of which we provide measurable evidences of the extent to which we succeed in meeting the very challenging objectives will include prototype implementations using open source application servers and demonstrators as well as technical reports and papers. Given the risky nature of the research project, precise metrics for measuring progress are difficult to formulate, other than the milestones that we have identified for each workpackage (see section 9.3). Indeed - there can be advantages in exploiting unforeseen new directions that are suggested by interim results and new software developments (particularly true for the world of Web services and Open Source software!). To help assess the progress we are making in the project, we have formed an Industrial Advisory Board. The role of the advisory board will be (i) to guide and validate our research; and (ii) to provide a means of dissemination of our results. The Board will meet at the end of each year and possibly at month six when the requirements deliverable (D1) becomes available. As a matter of fact, the members of the advisory board will, de facto, act as external evaluators of the progress of our work. The role of the Industrial Advisory Board is to assist the consortium:

- with evaluating the direction and progress of the Project; in particular evaluation and assessment reports, D12, D13 and D14 will be prepared in consultation with the Board; minutes of the Board meetings and feedback from the Board will be included in these reports;
- by providing access to possible further case studies;
- with the development of exploitation plans.

The membership of the Board is given in the Appendix (A2.6).

9.2.5 WP5: Dissemination and implementation

This workpackage aims at ensuring wide dissemination of the results of the project. In order to do so, members of the consortium will produce a dissemination and use plan and also a technological implementation plan. The former plan, the dissemination and use plan (deliverable report D16 produced at the beginning and its revised and updated version, D17, produced by the end of the second year) will not only envisage plans for dissemination of knowledge gained during the work, for example via presentation at technical conferences, participation in and/or organisation of specialist workshops, but also the exploitation plans of the results for the consortium as a whole and for individual participants or groups of participants. As stated earlier, software deliverables (D4, D8, D9 and D10) will be made available as open source software, thereby ensuring wide dissemination of the results of the project. The technological implementation plan will be written at the end of the project (deliverable report D18). It will describe the participants' actual achievements in dissemination and their plans at that time for the exploitation of their results.

9.2.6 WP6: Project Management

The TAPAS Executive Board will be responsible for the overall conduct of the project, setting overall technical goals, co-ordinating and reviewing technical progress, revising the project plan as necessary, ensuring the timeliness and quality of the deliverables, and developing an exploitation plan. It will be assisted by an Administrative Coordinator. Membership of the Executive Board will consist of one representative of each Partner.

The Executive Board will meet at least twice a year, normally in connection with Project workshops, and will also have regular meetings with the Advisory Board. Additional meetings will be called as and when necessary. Detailed responsibility for the different workpackages has been divided among the Partners - who nominate particular individuals to have detailed responsibility for the completion of the work and the timely production of the deliverables.

The Administrative Coordinator is, from the point of view of the Commission, the contract manager responsible for maintaining contact with the Commission on all aspects of the Project, and has primary responsibility for managing the budget, maintaining an information service, etc.

The project's infrastructure will use the Internet, and will consist of an internal and public web server, and an internal servers for documents, code and documentation, version control and archived mailing lists.

9.3 Workpackage descriptions

Workpackage number :	WP1- Application Service Requirements and Specification					
Start date or starting event:	Month 1					
Participant number:	P1	P2	P3	P4	P5	Total
Person-months per participant	12	11	14	24	7	68

Objectives:

To develop notations for expressing SLAs to enable specification of QoS, such as the availability, performance and scalability characteristics of components, as well as trust relationships. Model checking capabilities will be developed to support reasoning about QoS characteristics of components and their composition. This will support the ASP in assessing the qualities of service perceived by end-users prior to developing and deploying an application service.

Description of work:

This workpackage has been divided into four tasks. The first two tasks study, respectively application hosting requirements and networking requirements for many-to-many communications. These results will feed into WP2 and WP3. The remaining two tasks concern SLA specification and analysis: alongside functional components interfaces, these SLAs specify the non-functional qualities of service, such as the availability, performance and scalability characteristics of components; analysis tools will be developed for reasoning about the performance and scalability characteristics of components and their composition. TAPAS will adapt the UML as the baseline for the description, modelling and analysis of application services and extend the UML with formally defined stereotypes and properties that support the modelling of distributed interaction primitives and QoS characteristics. The semantics and properties of the stereotypes will be defined by mapping stereotyped UML diagrams to process algebra; in addition, we propose to use Markovian stochastic process algebra for use in performance analysis.

Deliverables:

Report D1 describing application hosting requirements and networking requirements

Report D2 will document the SLA specification method

Report D3 will describe service composition and analysis method

D4 will be open source software toolkit for service composition and analysis

The results from D2-D4 will be used for the development of applications in WP4.

Milestones and expected result:

Application hosting and network QoS requirements; service composition and analysis techniques. Results from D1 will feed into WP2 and WP3

Results from D2-D4 will be used for the development of applications in WP4.

Preliminary version of the toolkit available, month 19.

Workpackage number :	WP2- Design of QoS-aware Infrastructure for Application Hosting					
Start date or starting event:	Month 7					
Participant number:	P1	P2	P3	P4	P5	Total
Person-months per participant	24	11	14	12	7	68

Objectives:

To develop QoS enabled middleware services capable of meeting SLAs agreed between application services, and to enhance component based middleware technologies such that components can be deployed and interact across organisational boundaries. To consider hosting of important class of large scale networked *multi-party applications* that are characterised by a group of entities requiring many-to-many interactions; examples include collaborative applications, electronic share dealing and auctions, and 'information supply chain' applications typically based on publish-subscribe paradigm.

Description of work:

Component oriented middleware promotes the use of *containers* to host component instances. Containers are responsible for using the underlying middleware services for communication, persistence, transactions, security and so forth. TAPAS architecture will use SLAs not only as an inter-organizational contractual feature but also to govern component execution. To this end, QoS negotiation, establishment, and adaptation facilities will be added to the middleware and these will be used by component containers to make them QoS aware. The workpackage will have four tasks: middleware architecture, trust management, network control architecture and component execution environment. We are assuming that the organisations involved might not trust each other, so an important part of our work will be concerned with developing infrastructure support for interactions between two or more mutually distrusting organisations. Our aim is to ensure that all the non-functional properties will be derived from the SLAs and used to configure the execution environment at the time of component deployment.

Deliverables

Report D5: available end of the first year of the project, that will describe the interim TAPAS architecture for application hosting

Final architecture report, D6, describing the TAPAS architecture for application hosting

Milestones and expected result

Techniques for building QoS-aware containers

M09: Preliminary methods for governing QoS contracts between containers and services

M09: Understanding of the use of TTPs in inter-organisation interactions

M09: Signalling protocols for multi-party communications

M31: Feedback from case studies and evaluation work (WP4) available for D6

Workpackage number :	WP 3- Implementation of QoS-aware Core Services

Start date or starting event:	Month 7					
Participant number:	P1	P2	P3	P4	P5	Total
Person-months per participant	24	12	27	20	12	95

Objectives

Implement the architecture developed in WP2. Implement QoS-aware containers and support services of QoS-aware group communication, trusted coordination and QoS monitoring.

Description of work

Applications require the ability to request resources based on their immediate usage needs. Fulfilment of a given resource request requires several interactions with the underlying service, for which specific signalling methods and protocols (for end to end QoS negotiation) for resource reservation and feedback and adaptation will be implemented. QoS enabled Internet transport service based on PGM protocol will be developed. This will be used for supporting group communication and publish-subscribe protocols. For trusted coordination, implementation of two party client-server interaction with non-repudiation will be performed and extended to include TTPs. This framework will be further extended to support fair exchange protocols and multi-party communications. Component execution middleware API developed will be used to monitor conformance of component execution with the SLAs. TAPAS will make use of open source application servers (such as Jboss or Jonas) for this workpackage.

Deliverables

Deliverable report, D7 will define preliminary APIs for QoS-aware component containers

D8: prototype QoS-aware group communication system (open source)

D9: prototype trusted coordination software (open source)

D10: prototype QoS monitoring software (open source)

D11: Report, final APIs for QoS-aware component containers

Milestones and expected result

Open source application server software with QoS aware containers and services, permitting inter-organisation component deployment

M16: Preliminary method for implementing QoS contracts between containers and underlying services

M16: Preliminary version of enhanced PGM transport layer

M21: Preliminary versions of QoS-aware group communication, trusted coordination and monitoring software

Workpackage number :	WP 4- Evaluation and Assessment
Start date or starting event:	Month 7

Participant number:	P1	P2	P3	P4	P5	Total
Person-months per participant	11	20	12	11	6	60

Objectives

The TAPAS project will evaluate the methods, application servers and network protocols that it will develop by comparing them to current state-of-the-practice (e.g. an off-the-shelf CORBA or J2EE application server). Our principle way of evaluation will be to develop demonstrator applications using the facilities of the ASP partner. The project will port the components of these applications to the TAPAS platform and compare their execution characteristics to those developed with off-the-shelf products. It is our intention to demonstrate that the TAPAS platform can host applications that ASPs are currently unable to host.

Description of work

The variety of requirements for the application will be available from the requirements analysis performed in WP1 and will cover some basic topics such as (i) Trust management, (ii) Security policies, (iii) Privacy, and (iv) Measurable quality of the underlying network such as bandwidth, response time etc. The application, deployed and hosted on dedicated servers, will be usable for a group of testers. The installation should sufficiently demonstrate the results from the perspectives of the different participants such as end-users, service providers etc. For evaluation, it is necessary to test the fulfilment of the requirements under normal conditions as well as under stress situations. Due to the variety of requirements such as guaranteed bandwidth, simultaneous arrivals, transaction safety including different systems etc. constructing stress situations will lead to a variety of possibilities: (i) limiting the network bandwidth, (ii) causing significant delays of messages, (iii) trying to access protected data, (iv) stopping features like encryption in underlying services, (v) accessing services beyond the asserted limitations of data amount or accesses.

Deliverables

Views and feedback from the Industrial Advisory Board will be taken into account in the preparation of Evaluation and Assessment Reports.

D12: First Year Evaluation and Assessment Report

D13: Second Year Evaluation and Assessment Report

D14: Third Year Evaluation and Assessment Report

D15: QoS-aware and trusted ASP for Auctions, Demonstrator

Milestones and expected result

M24: preliminary version of auction system working

M31: preliminary evaluation results available

Results will provide inputs to TAPAS Architecture revision work

Workpackage number : WP 5- Dissemination and Implementation

Start date or starting event:	Month 1					
Participant number:	P1	P2	P3	P4	P5	Total
Person-months per participant	3	4	1	1	1	10

Objectives

The goal of this WP is to ensure dissemination of knowledge gained during the work and to propose exploitation plans for the results for the consortium as a whole, or for individual participants or groups of participants. In the course of this project, various options and strategies will be envisaged and discussed during regular Industrial Advisory Board meetings.

Description of work

The work carried out in this workpackage consists in the writing of two different plans. The first one, called the Dissemination and Use plan, will be produced at the very beginning of the project (D16). It will highlight the various strategies partners envisage to enable a wide dissemination of the expected results of the proposals. This plan will be updated, end of second year (D17). The second plan, called the Technological Implementation Plan, is the counterpart of the previous one, to be written at the end of the project. It will describe what participants have actually achieved to foster the dissemination of the now obtained results of the projects (D18)

Deliverables

D16: Dissemination and Use plan

D17: Updated Dissemination and Use plan

D18: Technological Implementation plan.

Milestones and expected result

M06: writing of D16 completed

M18: interim meeting to track each participant's achievements with respect to dissemination

M24: writing of D17 completed.

M36: writing of D18 completed.

Workpackage number :	WP 6- Project Management					
Start date or starting event:	Month 1					
Participant number:	P1	P2	P3	P4	P5	Total
Person-months per participant	43	4	4	4	3	58

Objectives

- To initialise, manage, and administer the project.
- To ensure the desired high quality and timely production of results.
- To set-up and maintain a communication infrastructure for the project.
- To provide administrative support for, and ensure that full benefit is gained from, the Industrial Advisory Board

Description of work

This work package is concerned with management at a project, activity, and partner level. The Executive Board, consisting of one representative of each partner, is responsible for the overall conduct of the project, including setting overall technical goals, co-ordinating and reviewing technical progress, revising the project plan as necessary, and ensuring the timeliness and quality of the deliverables. The Executive Board will be supported in this work by the Administrative Co-ordinator who will act as contract manager responsible for maintaining contact with the Commission, managing the budget, supporting the activities of the Industrial Advisory Board, ensuring proper dissemination of reports, etc.

WP6 will co-ordinate the production of the four technical Workpackage Reports and software development work.

Deliverables

D19: Project Presentation (T0 + 6 months)

D20: Final Report (T0 + 36 months)

PP1-3: Periodic Progress Reports, M12, M24 and M36

PM1-3: Periodic Management Reports, M06, M18, and M30

Milestones and expected result

M06: Project Presentation available on the web

M36: Final Report

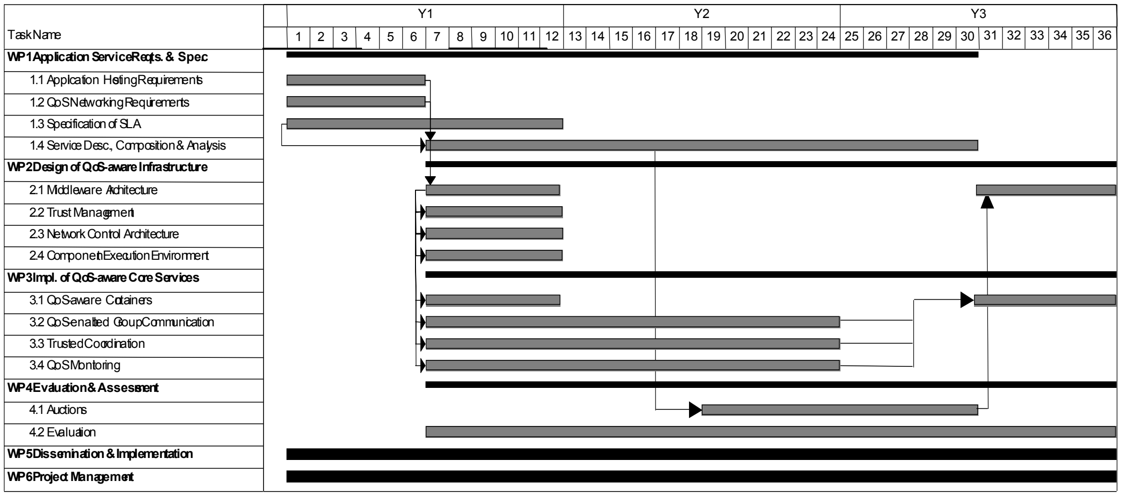
Realisation of regular technical and management project meetings per year

9.4 Deliverables list

Del. No.	Deliverable Name	WP No.	Lead participant	Estimate person-months	Del. type*	Security**	Delivery (proj. month)
D1	Application Hosting and Networking Req. Document	1	Adesso	10	R	Pub	6
D2	Specification Language for SLAs	1	UCL	10	R	Pub	12
D3	Method for Service Composition and Analysis	1	UCL	18	R	Pub	18
D4	Service Composition & Analysis Tool	1	UCL	30	P	Pub	30
D5	Architectural Design Document	2	UNEW	40	R	Pub	12
D6	Revised Architectural Design Document	2	UNEW	28	R	Pub	36
D7	QoS Container Interface Specification	3	UniBo	10	R	Pub	12
D8	Container for Group Communication	3	UCAM-CLAB	25	P	Pub	24
D9	Container for Trusted Coordination	3	UNEW	25	P	Pub	24
D10	Container for QoS Monitoring	3	UniBo	25	P	Pub	24
D11	Revised Container Interface Specification	3	UniBo	10	R	Pub	36
D12	First year Evaluation and Assessment Report	4	Adesso	5	R	Int	12
D13	Second Year Evaluation and Assessment Report	4	Adesso	5	R	Int	24
D14	Third Year Evaluation and Assessment Report	4	Adesso	10	R	Int	36
D15	QoS-aware and trusted ASP for Auctions	4	Adesso	40	D	Pub	30
D16	Dissemination and Use Plan	5	UNEW	3	R	Pub	6
D17	Updated Dissemination and Use Plan	5	UNEW	3	R	Pub	24
D18	Technological Implementation Plan	5	UNEW	4	R	Pub	36
D19	Project Presentation	6	UNEW	5	R	Pub	6
D20	Final Report	6	UNEW	13	R	Pub	36
PP1-3	Periodic Progress Reports		UNEW	20	R	Int	12, 24, 36
PM1-3	Periodic Mgt. Reports		UNEW	20	R	Int	6, 18, 30

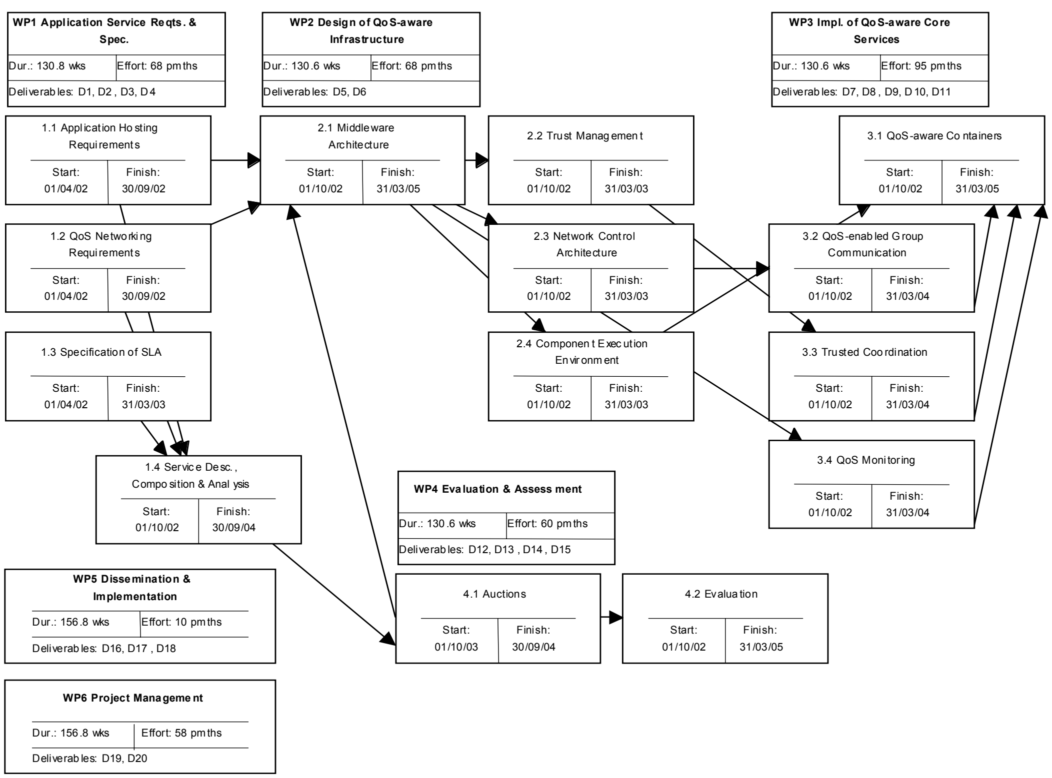
* *A short, self-evident description e.g. Report, Demonstration, Conference, Specification, Prototype…*
****Int.** *Internal circulation within project (and Commission Project Officer if requested)*
Pub. Public document

9.5 Project planning and timetable



9.6 Graphical presentation of project components

Note: Assumed project start date: April 2002



9.7 Project management

The Project Management of TAPAS is based on the experience and best practice developed by its members over many years of involvement in Esprit and other European Programmes. In particular, Newcastle University has 10 years experience as Co-ordinating Contractor of 15 major European collaborative projects and networks.

In the light of this experience, it is our intention to maintain as lightweight a management structure as possible within the constraints imposed by the need to deal effectively with intra- and inter-workpackage co-ordination and communication, and with the requirements detailed in the contract.

The project's infrastructure will use the Internet, and will consist of an internal and public web server, and of internal servers for documents, code and documentation, version control and archived mailing lists. The infrastructure of a previous project will be (re)used, as far as possible.

There will be two separate domains of management and co-ordination within TAPAS:

- The Executive Board that is responsible for the overall conduct of the project, and is the supreme decision-making body;
- The Administrative Co-ordinator who is responsible for providing day-to-day managerial assistance to the project.

Membership of the Executive Board will consist of one representative of each Partner. Where the Partner defines the role of principal investigator of a project, the principal investigator will normally be the representative on the Executive Board. The membership of the Executive Board will be: Prof. Santosh Shrivastava (UNEW), Dr. Wolfgang Emmerich (UCL), Prof. Fabio Panzieri (UNiBO), Prof. Volker Gruhn (Adesso) and Prof. Jon Crowcroft (UCAM-CLAB).

The Executive Board will be chaired by Prof. Santosh Shrivastava, principal investigator for Newcastle. The Board will have the primary responsibility of ensuring effective technical collaboration in pursuit of the overall aims of TAPAS.

The primary responsibilities of the Executive Board are as follows:

- to reach agreement between the Partners on relevant major technical issues
- to provide regular co-ordination and review of the technical workpackages, in close collaboration with the Industrial Advisory Board
- to approve the major project deliverables prior to submission to the EC
- to organise major project workshops and project reviews
- to support the Administrative Co-ordinator by providing the necessary authority on non-technical matters on behalf of each institution
- to deal with any non-technical issues between the Partners that cannot be resolved at the operational level by the Administrative Co-ordinator
- to co-ordinate the technical work of the project with that of other relevant IST projects
- to promote knowledge and understanding of TAPAS in circles external to the project (for example in industry, other research institutions, and academia)
- to resolve any conflicts that may arise during the course of the Project, following the Conflict Resolution Procedure described below.

The Executive Board will meet at least twice a year, normally in connection with Project workshops and will also have regular meetings with the Advisory Board. Additional meetings will be called as and when necessary. At each meeting of the Executive Board it will receive a report from the Administrative Coordinator (who will act as secretary to the Board). Day-to-day communication amongst members on Executive Board matters will normally be by means of an electronic bulletin board. As regards the day-to-day technical operation of the project, responsibility for the different components of the Workplan has been divided among the Partners. The organisation is as shown here:

WP1: Application Service Requirements and Specification (Wolfgang Emmerich - UCL)

WP2: Design of QoS-aware Infrastructure for Application Hosting (Santosh Shrivastava - UNEW)

WP3: Implementation of QoS-aware Core Services (Fabio Panzieri - UNiBO)

WP4: Case Studies and Evaluation (Volker Gruhn - Adesso)

WP5: Dissemination and Implementation (Santosh Shrivastava - UNEW)

WP6: Management and Co-ordination (Santosh Shrivastava - UNEW)

For each work item, the task/work package leader is responsible for the completion of the work and the timely production of the deliverables. Task leaders will report to the Executive Board. Newcastle will ensure the overall coordination of the Project.

The Administrative Co-ordinator is, from the point of view of the Commission, the contract manager responsible for maintaining contact with the Commission on all aspects of the Project. The Administrative Co-ordinator is primarily responsible for managing the budget; ensuring contractual deliverables are in fact delivered on time; ensuring the proper dissemination of internal and external technical (and other) reports; receiving from each site from time to time such technical, financial and administrative information as the Commission may require and forwarding it to them.

The Administrative Co-ordinator, who will be based at Newcastle, will carry out the following duties in a timely manner, under the direction of the Executive Board (through the principal investigator at the co-ordinating contractor site):

- develop and maintain expert knowledge of contractual matters that are administrative and financial in nature
- collect, monitor and integrate administrative and financial information from Project partners
- ensure the collection, storage and distribution of Project information, technical as well as administrative and financial
- convene the meetings of the Executive Board
- record the decisions taken by the Executive Board and monitor their implementation
- support the working of the Industrial Advisory Board
- ensure that information requested by the Commission is duly transmitted as and when required
- ensure the obligations of the Co-ordinator to the consortium as a whole are expedited (particularly with respect to financial and administrative matters)
- establish and maintain an efficient information service on the activities of the Project for use by members of the Project and interested external parties

Conflict Resolution Procedure: In normal operation, conflicts will be resolved in discussion with the appropriate Task Leader or, in the case of non-technical matters, the Administrative Coordinator. Where a conflict is not amenable to resolution through such discussions, the matter will be raised at the next scheduled Executive Board meeting or, if requested by any one of the Partners to the Project, at a specially convened meeting. Any conflict that is not amenable to resolution through discussion in Executive Board will be resolved by means of a vote. Each of the Partners to the Project will be eligible to cast a single vote. The decisions of the Executive Board, whether arrived at by discussion or vote, will be binding on all Partners to the Project. Finally, in addition to compliance with the standard model contract, it is our intention to produce a Consortium Agreement that will codify the specific rights and obligations of the Project Partners with respect to this particular project.

The project will be regularly reviewed by the Commission via review meetings and the involvement of external experts. Reviewing is an important activity of the project and the consortium will agree with the Commission, in due course, the precise mechanism and modality for carrying out such an activity.

10. Clustering

We will share ideas with existing IST projects and new projects in CPA12 on topics of mutual interests. Some of the existing projects include MAFTIA (fair exchange protocols, dependable trusted third parties), DSOS (system integration), CORTEX (realtime distributed systems).

11. Other contractual conditions

11.1 Travel

The project travel budget allows for participation in at least two project workshops per year by all those employed on the project and, in the case of "additional cost" partners, by members of permanent staff working on the project. It also allows for attendance at the yearly review. In addition, in order to provide visibility to the work carried out in the context of the project, the budget will allow members of the consortium to present TAPAS results at conferences, workshops and symposia in the fields of software engineering, middleware, dependability, reliability, security and related fields. Conferences, workshops and symposia that the consortium might participate in include:

Software Engineering Conferences: IEEE/ACM International Conference on Software Engineering, ACM SIGSOFT Foundations on Software Engineering, Automated Software Engineering Conference.

Middleware Conferences: IEEE/OMG International Enterprise Distributed Object Computing Conference, IEEE Intl. Symp. on Object-oriented Real-time distributed Computing, IFIP/ACM International Conference on Distributed Systems Platforms, IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems, USENIX Conference on Object-Oriented Technologies and Systems.

Dependable Computing Conferences: IEEE Symp. on Reliable Distributed Systems, IEEE/IFIP International Conference on Dependable Systems and Networks, Symposium on Security and Privacy, International Conference on Distributed Computing Systems.

The Coordination Travel Budget allows for attendance at the project workshops and review; the Other Specific Cost Budget will cover travel costs by the members of the Industrial Advisory Board (four meetings with six members).

Bibliography

[1] Rumbaugh, J., Jacobson, I. and Booch, G. The Unified Modeling Language Reference Manual. Addison Wesley. 1999

[2] Bernardo, M. Gorrieri, R. and Donatello, L. MPA: A Stochastic Process Algebra. Technical Report. UBLCS-94-10, Dept. of Computer Science, University of Bologna, 1994.

[3] Gilmore, S. and Hillston, J. The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. In Haring, G and Kotsis, G (eds). Proc. of the 7th Conf. On Modelling Techniques and Tools for Computer Performance Evaluation, 1994.

[4] Hillston, J. A Compositional Approach to Performance Modelling. PhD thesis. Dept. of Computer Science, University of Edinburgh, UK, 1994.

[5] Götz, N., Herzog, U. and Rettelbach, M. Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebras. In Proc. of the 16th Int. Symposium on Computer Performance Modelling, Measurement and Evaluation, PERFORMANCE 93. LNCS 729, Springer 1993.

[6] Klehmet, U. and Mertsiotakis, V. TIPPtool: Timed Processes and Performability Evaluation. User's Guide – Version 2.3, Technical Report 3/98, IMMD VII, University of Erlangen-Nürnberg, May 1998.

[7] http://www.jboss.org/

[8] http://www.evidian.com/jonas/

[9] Weiss, Walter: QoS with Differentiated Services. Bell Labs Technical Journal, December 1998.

[10] Clark, David D and Blumenthal, Marjory S: Rethinking the Design of the Internet: the End to End Arguments vs. the Brave New World. ACM Trans. On Internet Technology, 1,1, August 2001.

[11] Blaze, M, Feigenbaum, J. and Lacy, J: Decentralised trust management. Proc. Of IEEE Symp. On Security and Privacy, Oakland, May 1966.

[12] Froomkin, M.: The Essential Role of Trusted Third Parties in Electronic Commerce. Oregon Law Journal, vol. 49, 1996.

[13] N. Minsky and V. Ungureanu, "Law-Governed Interaction: A Coordination and Control Mechanism for Heterogeneous Distributed Systems", ACM Trans. Software Engineering and Methodology, vol. 9, pp. 273-305, July 2000.

[14] S. Ketchpel and H. Garcia-Molina, "A sound and complete algorithm for distributed commerce transactions", Journal of Distributed Computing, vol. 12, pp. 13-29, Jan. 1999.

[15] Louridas, P, "Some guidelines for non-repudiation protocols", ACM Computer Communications Review, October 2000

[16] Wichert, M. et al, "Non-repudiation evidence generation for CORBA objects using XML", Proc. of IEEE Computer Security Applications Conf., Arizona, Dec. 1999.

[17] Asokan, N, "Fairness in Electronic Commerce", Research Report RZ 3027, IBM Research, Zurich, May 1998.

[18] Shmatikov, V and Mitchell, J, "Analysis of a Fair Exchange Protocol", Proc. Internet Soc. Symp. On Network and Distributed System Security, 2000.

[19] Vogt, H. et al, "Modular Fair Exchange Protocols for Electronic Commerce", Proc. of IEEE Computer Security Applications Conf., Arizona, Dec. 1999.

[20] Kern, Thomas and Kreijger, Jeroen. "An Exploration of the Application Service Provision Outsourcing Option", Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS-34), January 3-6, 2001, Maui, Hawaii – Track 8. IEEE Computer Society, 2001, http://computer.org/proceedings/hicss/0981/volume%208/0981toc.htm

[21] Sun Microsystems, "Managing service levels for ASP success, A Guide to Delivering on Service Level Agreements", White Paper from http://www.sun.com/software/solutions/third-party/hosting/whitepapers/pdf/SunSLA_GD3.19.pdf

[22] Emmerich, W. and Kaveh, N. Model Checking Distributed Objects. In Proc. of the 4th Int. Software Architecture Workshop, Limerick, Ireland. ICSE Workshop Notes. May 2000.

[23] Hoare, C. A. R. Communicating Sequential Processes. Communications of the ACM 21(8): 666-677. 1978

[24] Milner, R. Communication and Concurrency. Prentice Hall. 1989

[25] Magee, J. and Kramer, J. Concurrency: From Finite State Processes to Java Programs. John Wiley & Sons. 1999

[26] Formal Systems Europe Ltd. Failures-Divergence-Refinement: FDR2 User Manual, 1997.

[27] Kaveh, N and Emmerich, W. Deadlock Detection in Distributed Object Systems. In V. Gruhn (ed) Joint Proc of the Joint 8th European Software Engineering Conference (ESEC) and the 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9), Vienna, Austria. pp. 44-51. ACM Press. 2001.

[28] T. Saridakis and V. Issarny. "Developing Dependable Software Systems using Software Architectures," in *Software Architecture,* Kluwer Academic Publisher, 1999.

[29] R. Balter et al., "Architecturing and configuring distributed applications with Olan", IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 1998), Springer-Verlag, 1998.

[30] Blair, G.S., "An architecture for next generation middleware", IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 1998), Springer-Verlag, 1998.

[31] H. Kopetz, "Component-Based Design of Large Distributed Real-Time Systems," *Control Engineering Practice - A Journal of IFAC, Pergamon Press*, vol. 6, pp.53-60, 1998.

[32] OMG, "CORBA components: joint revised submission", OMG TC Document orbos/99-07-01.

[33] Java™ 2 Platform, Enterprise Edition Specification Version 1.3. Copyright 1999-2000, Sun Microsystems, Inc. Available at http:// java.sun.com/j2ee/docs.html.

[34] Mervie, R and Merle, P, "CORBA Component Model: discussion and use with OpenCCM", Special Issue of the Informatica - An International Journal of Computing and Informatics (to appear).

[35] Bacon, J., K. Moody and W. Yao, "Access control and trust in the use of widely distributed services", IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), November 2001, Heidelberg.

[36] Grandison, T and Sloman, M, "A survey of trust in Internet applications", IEEE Communications Surveys, Fourth Quarter 2000, www.comsoc.org/pubs/surveys

[37] Brian Neil Levine, Jon Crowcroft, Christophe Diot, J. J. Garcia-Luna-Aceves and James F. Kurose, Consideration of Receiver Interest for IP Multicast Delivery, in Proc. of the Conference on Computer Communications (IEEE Infocom), (Tel Aviv, Israel), Mar. 2000.

[38] L. Vicisano, L. Rizzo and J. Crowcroft, TCP-Like Congestion Control for Layered Multicast Data Transfer, in Proceedings of the Conference on Computer Communications (IEEE Infocom), (San Francisco, California), March/April 1998.

[39] Blair. G. et al, "The design and implementation of Open ORB 2", IEEE Distributed Systems Online, October 2001, www.computer.org/dsonline/0106/

[40] M.-O. Killijian, J.-C. Fabre, J.-C. Ruiz-Garcia and S. Chiba. "A metaobject protocol for fault-tolerant CORBA applications," in *Proc. 17th Symp. on Reliable Distributed Systems (SRDS-17),* pp. 127-134, West Lafeyette, IN, USA, IEEE Computer Society Press, 1998.

[41] Daminanou, N, et al, "The Ponder policy specification language", Proc. Policy 2001 workshop, LNCS, 1995, January 2001.

[42] A. Romanovsky, J. Xu and B. Randell. "Exception Handling in Object-Oriented Real-Time Distributed Systems," in *Proc. 1st IEEE International Symposium on Object-Oriented Real-time Distributed Computing (ISORC'98),* pp. 32-42, Kyoto, Japan, 1998.

[43] R. Vanegas, J. Zinky, J. Loyall, D. Karr, R. Schantz, D. Bakken, "QuO's Runtime Support for QoS in Distributed Objects", Proc. IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware'98), The Lake District, New England, September 1998.

[44] Y. Krishnamurthy, V. Kachroo, D.A. Karr, C. Rodrigues, J.P. Loyall, R. Schantz,, D.C. Schmidt, "Integration of QoS-enabled Distributed Object Computing Middleware for Developing Next-generation Distributed Applications" Proc. ACM SIGPLAN Workshop on Optimization of Middleware and Distributed Systems (OM 2001), Snowbird, June 2001.

[45] B. Stiller, C. Class, M. Waldvogel, G. Caronni, D. Bauer, "A Flexible Middleware for Multimedia Communication: Design, Implementation and Experience", IEEE JSAC, Vol. 17, N. 9, 1580-1598, 1999.

[46] Baochun Li, “AGILOS: A Middleware Control Architecture for Application-Aware QoS Adaptations”, Doctoral Thesis, University of Illinois at Urbana-Champaign, 2000.

[47] K. Birman and T. Joseph, "Exploiting virtual synchrony in distributed systems", Proc. of 11th ACM Symposium on Operating System Principles, Austin, November 1987, pp. 123-138.

[48] F.B. Schneider, "Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial", ACM Computing Surveys, 22(4), December 1990, pp. 299-319.

[49] R. Guerraoui and A. Schiper, “Software-based replication for fault tolerance”, IEEE Computer, April 1997, pp. 68-74

[50] B. Kemme and G. Alonso, "A suite of replication protocols based on group communication primitives", Proc. 18th IEEE Intl. Conf. on Distributed Computing Systems, ICDCS’98, Amsterdam, May 1998, pp. 156-163.

[51] OMG, CORBA Notification service specification, June 2000.

[52] Java Message Service Specification, Version 1.o2b, August 2001.

[53] Liao, T, “Global information broadcast”, IEEE Internet Computing, August 2000.

[54] Carzaniga, A et al, “Design and evaluation of wide-area event notification service”, ACM Trans. On Computer Systems, 19, 3, August 2001.

[55] G. Morgan, S.K. Shrivastava, P.D. Ezhilchelvan and M.C. Little, “Design and Implementation of a CORBA Fault-tolerant Object Group Service”, Distributed Applications and Interoperable Systems, Ed. Lea Kutvonen, Hartmut Konig, Martti Tienari, Kluwer Academic Publishers, 1999, ISBN 0-7923-8527-6, pp. 361-374.

[56] G. Morgan and S.K. Shrivastava, “Implementing Flexible Object Group Invocation in Networked Systems”, Proceedings of the International Conference on Dependable Systems and Networks, New York, June, 2000

[57] G. Morgan and P.D. Ezhilchelvan, "Policies for using Replica Groups and their effectiveness over the Internet" Proc. 2nd International COST264 Workshop on Networked Group Communication (NGC 2000), Palo Alto, California, 2000.

[58] Speakman, T., et al, “PGM reliable transport protocol specification”, work in progress, April 2000, www.ietf.org/rfc/rfc2327.txt

[59] F Panzieri and S K Shrivastava, "On the provision of replicated Internet auction services”, IEEE Intl. Workshop on Electronic Commerce, WELCOM’99, Proc. of 18th IEEE Symp. on Reliable Distributed Systems, Lausanne, 19 October, 1999, pp. 390-395.

[60] P.D. Ezhilchelvan, S.K. Shrivastava and M.C. Little, “A model and architecture for conducting hierarchically structured auctions”, IEEE Intl. Symp. on Object-oriented Real-time distributed Computing, ISORC 01, May 2001, Magdeburg, Germany.

Appendix A - Consortium description

A.1 General description of the consortium

The TAPAS consortium brings together significant expertise from the application hosting, distributed computing and middleware, fault tolerance, software engineering, computer security and computer networking communities. All the partners will take part and contribute to all the workpackages and its tasks. Below we describe their principal responsibilities.

The Dept. of Computing Science at the University of Newcastle: The coordinating contractor for TAPAS, the Distributed Systems group conducts research on concepts, tools and techniques for constructing distributed fault-tolerant systems that make use of standard, commodity hardware and software components. Current work is focused on dependable workflow management for cross-organisation workflows, information sharing in virtual enterprise and wide area group communication systems. The group has built a number of major distributed software systems as CORBA middleware services. Newcastle will lead WP2 and will play active role in the development of the TAPAS architecture, tasks on QoS enabled group communication, trust management, trusted coordination and auction application development.

Adesso: Adesso AG is a German mid-range company offering IT consulting, software development and application hosting. Main clients are insurance companies and banks, for which the company analyses, designs and implements enterprise relevant applications based on component technologies such as Enterprise JavaBeans. The variety of hosted applications include applications in fields of telecommunication (B2B) and Internet portals for online communities and banks. Hosting of applications in areas of insurance companies and banks in mainly performed by the clients themselves, though the requirements and concepts are defined by the consultants and developers of Adesso AG. Adesso will lead WP 4 and will provide case studies, hosting facilities and undertake auction application development and evaluation work. Adesso will also play an active role in WP1 by providing requirements of application hosting.

The Dept. of Computer Science, University of Bologna: The research group is currently investigating a number research issues in the design of QoS-preserving, distributed middleware platforms. Specifically, these issues include: i) strategies for providing World Wide Web service users with adequate QoS. This activity involves investigating the design of middleware services that can meet effectively application-level (i.e., end-to-end) QoS requirements of Internet-based, latency-sensitive multimedia applications; and investigation on the use of group communication mechanisms to support replication in database systems. Bologna will lead WP3 and will contribute to all the activities concerned with QoS, in particular, QoS monitoring and QoS aware component containers, as well as work on network control architecture and auction development. Subcontractor of Bologna will be the Institute for Information Science and Technology (ISTI, see C7.3), based in Pisa. The key personnel of this Institute will provide Bologna and the TAPAS project with highly qualified expertise on analysis and evaluation of computer network architectures and protocols,.

The Dept. of Computer Science, University College London (UCL): The Software Systems Engineering Group is concerned with the development of large and complex software intensive systems. It focuses on: the real-world goals for, services provided by, and constraints on such systems; the precise specification of system structure and behaviour, and the implementation of these specifications. The three key technologies where the group contributes to the state-of-the-art are: databases, distributed objects (particularly middleware and mobile agent technologies), web infrastructure (particularly XML and related technologies). UCL will lead WP1 and will contribute to the development of SLA specification and analysis tool, architecture development, QoS aware containers and evaluation.

Computer Laboratory, University of Cambridge: The Systems Research Group at the Computer Laboratory, University of Cambridge, has been one of the premier research forces in communications, distributed systems and operating systems since the founding of the lab, the oldest computer science teaching department in the world. Past projects include Universe (which delivered one of the earliest high speed distributed systems), nemesis, a novel operating system with excellent multimedia scheduling properties, as well as Home Area Networks, Xenos (an accountable peer-to-peer distributed architecture). Networks and Operating Systems related work is focusing on Disk QoS Enforcing Quality of Service in Storage Systems , Efficient Network Routeing, Next Generation Inter-AS Routeing. Cambridge will concentrate on QoS networking requirements (WP1), Network Control Architectures (WP2) and QoS enabled group communication (WP3).

A.2. Description of the Participants

A2.1 University of Newcastle, Department of Computing Science (UK)

The Department of Computing Science of the University of Newcastle upon Tyne is one of the largest in the UK. It has been awarding PhD degrees since 1965, and was the first department of computing science in the UK to gain Science and Engineering Research Council approval for an advanced MSc course. There are 27 academic teaching staff, 18 postgraduate research staff and approximately 90 postgraduate students.

The Department conducts research in a wide range of topics that reflect a broad systems view of the discipline - primarily concerning software, architectural, communications and applications issues. Emphasis is placed on combining theory and practice to develop new understanding and techniques and to provide effective solutions to problems of design, implementation and application in these areas. The Department has been awarded a Grade 5, or equivalent, rating in each of the four National Research Assessment Exercises. This rating indicates international excellence in the Department's main research areas: Dependability, Distributed Systems, Parallelism, Theoretical Computing Science, and VLSI Design. It is currently involved in over 30 research contracts, variously funded by UK EPSRC, DTI, EU and industry, with an annual income of approximately £1.5M. Industrial contracts include the BAe-sponsored Dependable Computing Systems Centre, which is a joint activity with the University of York; and projects with British Telecom, Hewlett Packard, Nortel Networks and Marconi Communications.

The Department has a strong position in the European Computing Science scene. It has led a number of major ESPRIT projects and networks in dependable and distributed computing systems since 1989, including the PDCS, the C3DS Esprit LTR Project on control and coordination of complex distributed services, the BROADCAST Working Group on large scale distributed computing and the CaberNet Network of Excellence on distributed computing systems architecture. It is coordinating two CPA 2 projects on dependability: MAFTIA and DSOS.

Santosh Shrivastava: Santosh Shrivastava was appointed a Professor of Computing Science, University of Newcastle upon Tyne in 1986; he leads the Distributed Systems Research Group. He received his Ph.D. in computing science from Cambridge in 1975. His research interests are in the areas of distributed systems, fault tolerance and application of transaction and workflow technologies to e-commerce. He led ESPRIT basic research project BROADCAST and long term research project C3DS on service provisioning. His group collaborates and receives funding from Nortel Networks, British Telecom, Hewlett-Packard and IBM. Together with his colleagues he set up Arjuna Solutions Ltd in 1998 in Newcastle to productise transaction and workflow technologies developed by his group. Now part of Hewlett-Packard, HP-Arjuna Labs (www.arjuna.com) is a centre of excellence for transactional technology.

Paul Ezhilchelvan: Paul Devadoss Ezhilchelvan received Ph.D. degree in computer science in 1989 from the university of Newcastle upon Tyne, United Kingdom. He received the Bachelor of Engineering degree in 1981from the University of Madras, India, and the Master of Engineering degree in 1983 from the Indian Institute of Science, Bangalore. He joined the Department of Computing Science of the University of Newcastle upon Tyne in 1983 where he is currently a lecturer. His main research interests are in the areas of fault-tolerance and distributed computing. He has published several research papers in the topics of distributed agreement protocols, replicated processing, scalable and reliable multicast protocols, and group membership in synchronous and asynchronous distributed systems.

Graham Morgan: Graham Morgan is a lecturer in the Department of Computing Science. He received his PhD in Computing from Newcastle University in 2000. He was appointed to a Lecturer position on September 2000 and is a member of the Distributed Systems Research Group. His research interests are in the areas of distributed computing, group communications, fault-tolerant computing, real-time systems and Middleware software development. Recently, he has worked towards the development of protocols provided as Middleware services for satisfying real-time and consistency requirements for Internet based Auction applications and Networked Virtual Environments.

Nick Cook: Nick Cook joined the Department as administrative coordinator for research in 1989. As administrator on a succession of major ESPRIT-funded research projects he has maintained strong links with the European dependable and distributed systems research community. In September 1998, Nick began studying part-time in the Department for an MSc in Computing Science, which he completed in 2000. He has a BSc in Mathematics from University of Manchester (1982). Nick is a doctoral student now, working on infrastructure support for inter-organisation information sharing.

Selected Publications:

M.C. Little and S K Shrivastava, “Java Transactions for the Internet”, Distributed Systems Engineering, 5 (4), December 1998, pp. 156-167.

G. Morgan, S.K. Shrivastava, P.D. Ezhilchelvan and M.C. Little, “Design and Implementation of a CORBA Fault-tolerant Object Group Service”, Distributed Applications and Interoperable Systems, Ed. Lea Kutvonen, Hartmut Konig, Martti Tienari, Kluwer Academic Publishers, 1999.

D. Ingham, F. Panzieri and S.K. Shrivastava, “Constructing dependable Web services”, IEEE Internet Computing, Jan/Feb 2000, pp. 25-33.

G. Morgan and S.K. Shrivastava, “Implementing Flexible Object Group Invocation in Networked Systems’, IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-2000), June 2000, New York, pp. 439-448.

G. Morgan and P.D. Ezhilchelvan, "Policies for using Replica Groups and their effectiveness over the Internet" Proc. 2nd International COST264 Workshop on Networked Group Communication (NGC 2000), Palo Alto, California, 2000.

S.K. Shrivastava, L. Bellissard, D. Féliot, M. Herrmann, N. dePalma, S.M. Wheater, “A Workflow and Agent based Platform for Service Provisioning”, Proc. of 4th IEEE/OMG International Enterprise Distributed Object Computing Conference (EDOC 2000), September 2000, Makuhari, pp. 38-47.

P.D. Ezhilchelvan, S.K. Shrivastava and M.C. Little, “A model and architecture for conducting hierarchically structured auctions”, IEEE Intl. Symp. on Object-oriented Real-time distributed Computing, ISORC 01, May 2001, Magdeburg, Germany.

I. Houston, M. C. Little, I. Robinson, S. K. Shrivastava and S. M. Wheater, “The CORBA Activity Service Framework for Supporting Extended Transactions”, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), November 2001, Heidelberg.

A2.2 Adesso AG (D)

Adesso AG is a software house specialized in component-based software engineering. Most systems developed by Adesso are e-business applications. Adesso employs about 140 people in Munich, Frankfurt, Berlin, Cologne, and Dortmund. Main customers of Adesso are from the insurance, banking, and telecommunications area. The selling proposition of Adesso is to provide full-fledged support in the design, implementation and operation of e-business application. ASP services range from pure hardware operation to full service e-business controlling services (e.g., monitoring of customer navigation behaviour, click rate measurement, etc.). The variety of hosted applications includes applications in fields of telecommunication (B2B) and Internet portals for online communities and banks. Hosting of applications in areas of insurance companies and banks is mainly performed by the clients themselves, though the requirements and concepts are defined by the consultants and developers of Adesso AG.

As modern organisations become global and distributed, Adesso strongly believe that one-site hosting solutions will be replaced by network structures. Especially in modern Internet applications like broker portals or marketplaces the distribution of information sources and sinks seems inevitable as different companies have to interact.

Volker Gruhn: Professor Gruhn is chairman of the supervisory board and co-founder of Adesso. He is also a professor at the computer science department of the University of Dortmund. He is head of research group of 7 PhD students and research assistants. The focus of this group is the support of software processes in distributed and highly evolution-based circumstances. From 1994 to 1996 Volker Gruhn was chief technical officer at LION, a medium sized software house with 400 employees. In this position he was responsible for software development, quality management and the mainframe computing centre of LION. He was chief designer of the workflow management environment LEU and he was responsible for several software systems developed with LEU. The most important of these systems is a system supporting all business processes from the area of housing construction and administration. This system, called WIS, is one of the three leading systems in the German market. The development budget for WIS was about 16 million ECU.

Werner Beckmann: Werner Beckmann received a diploma degree from department of computer science of the University of Dortmund in 1992. Since then he participated in several software development projects. Since 1997 Werner Beckmann managed several development projects. Customers of these projects were Internet Service Providers, international training and education companies. His current interests are in distributed EJB architectures and business models for their operation. Werner Beckman is team leader of an EJB development team. As such he is responsible for development teams, who develop EJB-based applications for insurance companies and banks. The focus of these teams is to develop applications which can be operated by Adesso in an ASP business model.

Selected Publications:

W. Deiters, V. Gruhn, Process Management in Practice - Applying the FUNSOFT Net Approach to Large Scale Processes, in: Special Issue on Process Technology / Automated Software Engineering, 1998.

V. Gruhn, Software Process Landscaping, in: Software Process Improvement and Practice Journal, Volume 5, September 2000.

V. Gruhn, U. Wellen, Analyzing a Process Landscape by Simulation, in: Journal of Systems and Software, Vol. 59, No. 3, 2001.

V. Gruhn, S. Lembke, Flexible Integration of Petri Net Based Process Descriptions With User-Specific Data Descriptions, Journal of Integrated Design and Process Science, Society for Design and Process Science, Vol. 5, No. 1.

V. Gruhn, D. Peters, Concurrent Security Modelling in a Distributed Java-based E-Commerce Environment, in: U. Baake, J. Herbst, S. Schwarz (eds.), 8th European Concurrent Engineering Conference (ECEC 2001), April 2001, Valencia, Spain.

R. Balzer, V. Gruhn, Process-Centered Software Engineering Environments: Academic and Industrial Perspectives, in: Proceedings of the 23rd International Conference on Software Engineering, May 2001, Toronto, Canada.

S. Beydeda, V. Gruhn, Decision and Risk Analysis for Process Evolution, in: Proceedings of the Workshop on Software Process and Product Improvement at the EUROMICRO'2001 Conference, September 2001, Warsaw, Poland.

S. Beydeda, V. Gruhn, Integrating White- and Black-Box Techniques for Class-Level Regression Testing, in: Proceedings of the 25th Computer Software and Applications Conference (COMPSAC 2001), October 2001, Chicago, USA.

A2.3 University of Bologna, Department of Computer Science

The University of Bologna, founded in 1088, is the oldest university in Europe and one of the most highly regarded academic institutions in Italy. Computer Science teaching and research is carried out in the Department Computer Science, which was established in 1995, and include 9 full professors, 5 associate professors and 10 research associates. Starting with the 1994-95 academic year, the Department has initiated a 4-year graduate program leading to a PhD degree in addition to the existing 5-year undergraduate program. The principal research areas within the Laboratory include distributed systems, fault tolerance, real-time and multimedia systems, performance and dependability evaluation, distributed simulation, security, languages for coordination, formal specification of software systems, semantics of concurrency, type theory, logics and foundations of computation. Laboratory members are actively involved in several national and European research projects and CaberNet (Network of Excellence).

Subcontractor of Bologna will be the Institute for Information Science and Technology (ISTI); established in September 2000 in the frame of the global reorganization of the research institutes of National Research Council (CNR). ISTI was created, by merging the Institute for Information Processing (IEI) with CNUCE (CNUCE). Both institutes, located in the CNR Research Campus in Pisa, are internationally known and have decidedly contributed to the establishment and the development of Information Technology activities in Italy and at an international level. **Marco Conti** senior researcher and **Enrico Gregori** CNR research director will bring highly qualified expertise on analysis and evaluation of computer network architectures and protocols, and middleware services.

Fabio Panzieri: Fabio Panzieri is a professor of Computer Science at the Faculty of Science of the University of Bologna (Italy). He obtained the "Laurea" degree in "Scienze dell' Informazione" from the University of Pisa (Italy), in February 1978, and the Ph. D. degree in Computer Science from the University of Newcastle upon Tyne (U.K.), in June 1985. From June 1979 to May 1985 he was a research associate at the Computing Laboratory of the University of Newcastle upon Tyne. During these years he carried out research on fault tolerance issues in distributed systems, and on the design and development of communication protocols for local and wide area networks. From October 1985 to April 1989, he was an independent consultant for a number of medium and large IT companies. From April 1989 to October 1990 he was an associate professor at the Dipartimento di Matematica Pura ed Applicata of the University of L'Aquila (Italy). His research activity is concerned with issues of design of fault tolerant distributed systems, distributed real-time systems, and middleware and communication support for responsive distributed applications.

Ozalp Babaoglu: Ozalp Babaoglu is Professor of Computer Science at the University of Bologna, Italy. He received a Ph.D. in 1981 from the University of California at Berkeley where he was one of the principal designers of BSD Unix. Before moving to Bologna in 1988, Babaoglu was an Associate Professor in the Department of Computer Science at Cornell University. He is active in several European research projects exploring issues related to fault tolerance and large scale in distributed systems. Babaoglu serves on the editorial boards for ACM Transactions on Computer Systems and ACM Springer-Verlag Distributed Computing.

Renzo Davoli: Renzo Davoli is an Associate Professor of Computer Science at the Department of Computer Science of the University of Bologna. He received his degree in Mathematics from the University of Bologna (Italy) in 1986. In 1991 he joined the Department of Mathematics of the same University as a Research Associate. He has been a member of the Computer Science Department since 1995. His research interests include large-scale distributed systems, real-time systems, nomadic computing, wireless systems and neural networks.

Marco Roccetti: Marco Roccetti is a Professor of Computer Science at the Department of Computer Science of the University of Bologna. He received the Italian Laurea degree in Electronic Engineering from the University of Bologna in the academic year 1987/88. From 1990 to 1992 he was a member of the technical staff of the Department of Mathematics at the University of Bologna, and from December 1992 to October 1998 he was with the Department of Computer Science as a Research Associate. His research interests include: Design, Implementation and Evaluation of Multimedia Computing and Communication Systems, Performance and Reliability Analysis of Distributed and Parallel Computing Systems, Simulative and Probabilistic Methods for the Development of Distributed Computing Systems and Applications.

Selected Publications:

Panzieri, M. Roccetti, V. Ghini, "The Implementation of Middleware Services for QoS-aware Distributed Multimedia Applications", Proc. ACM Multimedia 2001, Ottawa, Canada, October 2001.

M. Conti, E. Gregori, F. Panzieri, "QoS-based Architectures for Geographically Replicated Web Servers", Cluster Computing 4, 2001, pp. 105-116, Kluwer Academic Publishers.

V. Ghini, F. Panzieri, M. Roccetti, "Client-centered Load Distribution: A Mechanism for Constructing Responsive Web Services", in IEEE Proc. 34th Hawaii International Conference On System Sciences (HICSS-34), Maui, Hawaii, January 3-6, 2001.

D. Ingham, S.K. Shrivastava, F. Panzieri, "Constructing Dependable Web Services", IEEE Internet Computing, Vol. 4, N. 1, January/February 2000, pp. 25 - 33.

M. Conti, E. Gregori, F. Panzieri, "Load Distribution among Replicated Web Servers: A QoS-based approach", Proc. 2nd ACM Workshop on Internet Server Performance (WISP'99), Atlanta Georgia, USA, May 1st, 1999.

B. Kemme, A. Bartoli, O. Babaoglu. Online Reconfiguration in Replicated Databases Based on Group Communication. Proceedings of the International Conference on Dependable Systems and Networks (DSN 2001), pages 117--126, Goteborg, Sweden, July 2001.

O. Babaoglu, R. Davoli and A. Montresor. Group Communication in Partitionable Systems: Specification and Algorithms. In IEEE Transactions on Software Engineering, 27(4):308-336, April 2001.

A. Montresor, R. Davoli, O. Babaoglu. Enhancing Jini with Group Communication. Technical Report UBLCS 2000-16, December 2000 (Revised January 2001), In Proceedings of the ICDCS Workshop on Applied Reliable Group Communication (WARGC 2001), April 2001, Phoenix, Arizona (USA).

A2.4 University College London, Department of Computer Science

University College is both the oldest and the largest college of the University of London. UCL has over five thousand undergraduates and two thousand postgraduates in over sixty departments. The Department of Computer Science was founded in 1973. It has long been a centre of excellence for computer networking, multi-media, virtual reality and more recently for software systems engineering. The Department was the first node on the Arpanet outside the US and has regained its excellence in networking. The department has approximately 35 academic staff, 45 full-time research staff and about 50 PhD students. A major part of the Department's research is applied, driven the through applications, user studies and pilots.

UCL's Software Systems Engineering Research Group, which is participating in TAPAS has a proven track record in software and systems engineering. Its main interests are in methods, tools and processes for software and systems requirements and architecture. The group's research agenda is focussed on applied and exploratory research and development. It maintains a lot of collaborations with the national and international industry.

The group acted as co-ordinator of the RENOIR ESPRIT-IV network of excellence. The group has participated in PROMOTER.2 on process modelling techniques and is now involved in the IST-funded project on Ubiquitous Web Applications (UWA). The Group has several bi-lateral research projects with British SMEs that are funded by the Teaching Company Directorate, a joint venture of the British EPSRC, the ESRC and the Department of Trade and Industry. One of these schemes is co-sponsored by Searchspace Limited and has resulted in the definition of the J2EE based architecture for application services for intelligent transaction monitoring.

The group has strong consultancy links with the National Air Traffic Services (NATS) for which a system requirement engineering process assessment was conducted and a requirements engineering standard was established. During the last year, the group has focussed attention on the qualitative and quantitative analysis of distributed object and component systems using process algebra and on application services provision.

Wolfgang Emmerich: Wolfgang Emmerich received his MSc from University of Dortmund, Germany in 1990 and his PhD from University of Paderborn in 1995. His PhD thesis was on database support for integrated software engineering environments. Wolfgang was a research assistant in the Dept. of Computer Science at Dortmund between 1990 and 1995. After his PhD he joined City University, London as a Lecturer and developed an interest in software engineering for distributed object-based systems. In November 1997, he joined UCL where he now has the position of a Senior Lecturer. His research interests are in application service provision, design of distributed object and component-based software architectures and mobile systems. Wolfgang was a senior consultant at the OMG Representative for Central Europe, where he developed his distributed object consulting expertise. He has become a recognised expert in the area of software engineering for distributed objects and is the author of a text book on "Engineering Distributed Objects" published by John Wiley & Sons. Wolfgang has extensively consulted in the European Software Engineering Industry and is now a Partner of the Zuhlke Technology Group and a Senior Consultant and Director of Zuhlke Engineering (UK) Ltd in London.

Cecilia Mascolo: Cecilia Mascolo holds a Laurea degree in Science dell' Informazione and a PhD in Informatica from the University of Bologna, Italy. In 1999, she spent a year as a Visiting Academic at the Department of Computer Science at Washington University, Saint Louis. In February 2000, she became a Research Fellow at the Department of Computer Science at University College London and joined the academic staff of the Department as a Lecturer in Computer Science in February 2001. Cecilia's research interests are in the area of software architecture for distributed and mobile systems. Cecilia brings a strong background in formal methods and the application of model checking techniques to software architectures into the TAPAS project.

Selected Publications:

W. Emmerich. OMG/CORBA: An Object-Oriented Middleware. In John J. Marciniak (ed): Encyclopedia of Software Engineering. John Wiley & Sons. 2001.

W. Emmerich and N. Kaveh. Component Technologies: Java Beans, COM, CORBA, RMI, EJB and the CORBA Component Model. In V. Gruhn (ed). Proc. of the Joint 8th European Software Engineering Conference and 9th ACM SIGSOFT Int. Symposium on Foundations of Software Engineering, Vienna, Austria. pp. 311-312. ACM Press. 2001.

G. Piccinelli, W. Emmerich and A. Finkelstein Mapping Service Components to EJB Business Objects. Proc. of the 5th Int. Enterprise Distributed Object Computing Conference (EDOC 2001). pp. 169-173. IEEE Computer Society Press. 2001.

N. Kaveh and W. Emmerich. Deadlock Detection in Distributed Object Systems. In V. Gruhn (ed). Proc. of the Joint 8th European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9), Vienna, Austria. pp. 44-51. ACM Press. 2001

P. Ciancarini, F. Franze and C. Mascolo. Using a Coordination Language to Specify and Analyze Systems containing Mobile Components. In ACM Transactions on Software Engineering and Methodology, 9(2). April 2000.

K. Takahashi, W. Emmerich, A. Finkelstein and S. Guerra. System Development using Application Services over the Net. In Proc. of the 22nd Int. Conference on Software Engineering, Limerick, Ireland. ACM Press. 2000.

W. Emmerich. Engineering Distributed Objects. Wiley. April 2000.

A2.5 Computer Laboratory, Cambridge University

Systems Research Group is the largest research area in the Computer Laboratory covering hardware, communications hardware and software, operating systems and distributed systems. Past systems developed here include Edsac, the Titan operating system, the Cambridge Ring and the Cambridge Distributed Computing System. The Systems Research Group (SRG) has some 40 members, and is divided into a number of subgroups: the Opera group, the Networks and Operating Systems group, and the Self-Timed Logic group. The first two are most relevant here. The thrust of the Opera Group is open distributed processing with an object orientated approach. We look to support emerging distributed applications in which users and devices may be mobile and may have multimedia presentation requirements. Current themes are asynchronous (event) extensions to synchronous middleware and an access control architecture, Oasis. Networks and Operating Systems group's relevant sub-projects involve Disk QoS Enforcing Quality of Service in Storage Systems, Efficient Network Routeing, Next Generation Inter-AS Routeing, Xenoservers - Accountable Execution Of Untrusted Program.

Jon Crowcroft is just moving from being a professor of networked systems in the Department of Computer Science University College London, to being the Marconi Professor of Communications Systems at the Computer Lab, Cambridge University. Jon's recent/current projects at UCL include the BT funded active networks project ALPINE, and the Learnet Infrastructure as well as helping the funded Internet 2 projects on ipv6 the Bermuda triangle, a DARPA funded project on radio-active networks, the EU FV Traffic Engineering for the Internet (Tequila), FV Federated Organisations Management (Form). Sprint, Nortel (UK and Canada) fund some Internet Telephony and IETF activities. Jon is a member of the ACM, a member the British Computer Society, a fellow of the IEE and the royal academy of engineering and a Senior Member the IEEE. He is also on the editorial team for Computer Networks, IEEE/ACM Transactions on Networking, IEEE Networks, Monet , and Cluster Computing. Jon is on the Internet Architecture Board.

Selected Publications:

Bharat Patel and Jon Crowcroft, Ticket Based Service Access for the Mobile User, ACM Mobicom 97, Budapest, 1997.

L. Vicisano, L. Rizzo and J. Crowcroft, TCP-Like Congestion Control for Layered Multicast Data Transfer, in Proceedings of the Conference on Computer Communications (IEEE Infocom), (San Francisco, California), March/April 1998.

J.Crowcroft, I.Kouvelas, V.Hardman ,Self Organising Transcoders, NOSSDAV, 1998.

Philippe Oechslin, J Crowcroft, Weighted Proportional Fairness and Pricing for TCP ACM CCR, Volume 28, Number 3 (July 1998).

J. Crowcroft, Herding Cats - Modelling the Internet, Royal Society, Philosophical Transactions, 1999.

Colin Perkins and Jon Crowcroft, Effects of Interleaving on RTP Header Compression, in Proceedings of the Conference on Computer Communications (IEEE Infocom), (Tel Aviv, Israel), Mar. 2000.

Brian Neil Levine, Jon Crowcroft, Christophe Diot, J. J. Garcia-Luna-Aceves and James F. Kurose, Consideration of Receiver Interest for IP Multicast Delivery, in Proc. of the Conference on Computer Communications (IEEE Infocom), (Tel Aviv, Israel), Mar. 2000.

Ghosh, Fry & Crowcroft, An Architecture for Application Layer Routing, in Yasuda, H. (Ed), Active Networks, LNCS 1942, Springer, pp 71-86. ISBN 3-540-41179-8 Springer-Verlag.

A2.6 TAPAS Industrial Advisory Board

The project will form an Industrial Advisory Board, whose membership will represent a cross-section of technology providers, end-users and middleware standards bodies. Regular meetings with the Board will help us in revising, where necessary, the objectives of the project. The membership of the Board includes:

Paul McKee (*BT exact Technologies*): is a team leader in the Distributed Computing and Information Systems research group at BT exact Technologies. He currently manages projects including collaboration with a number of Universities. His research is focused on large-scale distributed systems, particularly policy-based management and high performance event-based architectures for capturing and processing management information. Paul joined BT in 1989 and initially worked on high-resolution optical devices before moving to a distributed systems group where he worked on autonomous replication and low overhead consistency protocols. He has published over 40 papers and is a member of the IEEE Computer Society.

Andrew Watson (*Technical Director of the OMG*): graduated from the University of Cambridge in Computer Science and Engineering and spent two years at Hewlett-Packard's Bristol Research Centre, working on one of the first X.400 implementation. In 1989 Andrew joined the ANSA core team, working initially on the of the ANSA Computational Model and DPL, a language realising that model. Andrew then joined the Object Management Group (OMG) and chaired the ORB2 Task Force. Andrew is now Technical Director of the OMG and is responsible for the OMG's technology adoption process. Andrew also chairs the OMG's Architecture Board, a group of distinguished technical contributors from OMG member organizations. It was during Andrew's technical directorship that the OMG adopted the Unified Modelling Language (UML), the Common Object Request Broker Architecture (CORBA) and the CORBA Component Model.

Prof. Dr. Rudolf K. Keller (*Zühlke Engineering AG*): is the leader of the business unit Java Computing at Zühlke Engineering AG in Schlieren (Zürich), Switzerland. He is was an Associate Professor in the Software Engineering Group (GÉLO) at the Department of Computer Science and Operations Research at University of Montreal (UdeM). Before joining the faculty at UdeM in 1994, he was for several years a researcher at Montreal's CRIM research institute. Rudolf has taught at the School of Computer Science at McGill University and at University of California at Irvine, where he was a postdoctoral fellow from 1989 to 1991. He received a M.Sc. degree in mathematics from the Swiss Federal Institute of Technology (ETH) at Zürich in 1983, and a Ph.D. degree in computer science from University of Zürich in 1989. Rudolf's current interests are in object-oriented analysis and design, reverse engineering, design components and patterns, software quality, user interface engineering, business process modelling, and technologies for electronic marketplaces.

Dr. Marko Boger (*CEO of Gentleware AG*): is founder and CEO of Gentleware AG, a German company building UML-CASE-tools. He holds a PhD from the University of Hamburg where he worked as researcher on topics like UML, distributed systems development and e-Commerce for several years. He is author of the book 'Java in Distributed Systems', originally published in German (dpunkt-verlag) and later translated to English and published by Wiley. Marko was a key contributor to Argo/UML developer, which has now been developed by Gentleware into the Poseidon Toolsuite that is becoming part of Sun's Forte for Java development environment. Marko is a regular speaker at conferences, member of the program committee of the UML conference series and actively engaged in the standardisation of UML at the OMG.

Dr. Mark Little (*HP Arjuna Labs*): is a Distinguished Engineer/Architect, within HP Arjuna Labs., Newcastle upon Tyne, England, where he leads the Transactions team. He joined HP via a series of company acquisitions: Bluestone Software, Arjuna Solutions, which he was one of the founders. Before joining Arjuna Solutions he was for over 10 years a member of the Arjuna Distributed Computing team within the University of Newcastle upon Tyne (where he continues to have a Visiting Fellowship). His research within the Arjuna team included replication and transactions support, which include the construction of an OTS/JTS compliant transaction processing system.

Dr. Stuart Wheater (*HP Arjuna Labs*): is a Distinguished Engineer/Architect, within HP Arjuna Labs., Newcastle upon Tyne, England. He joined HP via a series of company acquisitions: Bluestone Software, Arjuna Solutions, which he was one of the founders. Before joining Arjuna Solutions he was for over 10 years a member of the Arjuna Distributed Computing team within the University of Newcastle upon Tyne (where he continues to have a Visiting Fellowship). His research within the Arjuna team included transactions and long-lived process support, which include the construction of a CORBA based transactional workflow system.

Dr. Tobias C. Kiefer (*Head of eTransaction Banking, Commerz NetBusiness AG/ Commerzbank Group*): Since April 2001 Head of eTransaction Banking at Commerz NetBusiness AG. Responsible for business development concerning epayments, mpayments, electronic bill presentment and payment, internet trust services and innovative transaction technologies and methods. Author of numerous publications and conference presentations concering the topic of services based on PKI, eBusiness strategies, banking strategies as well as speaker and moderator of specialized conferences with regard to strategies in e-commerce and etransaction banking. Main expertise in strategies, innovation management and business development.