INTEGRATING MOBILE AGENT INFRASTRUCTURES IN OPERATIONAL ERP SYSTEMS

Apostolos Vontas, Philippos Koutsakas, Christina Athanasopoulou, Adamantios Koumpis, Panos Hatzaras
ALTEC S.A., Software Business Unit, Research Programmes Division, Mich.Kalou 6, GR - 546 29, Thessaloniki, Greece
Email: avontas@unisoft.gr, fko@unisoft.gr, cat@unisoft.gr, akoumpis@unisoft.gr, phatzaras@unisoft.gr

Yannis Manolopoulos, Michael Vassilakopoulos
Data Engineering Laboratory, Department of Informatics, Aristotle University of Thessaloniki, GR 54006 GreeceEmail:
Email: manolopo@skyblue.csd.auth.gr, mvass@computer.org

Key words: Mobile Agents, distributed application environments, fractal company.

Abstract: In this paper we present our most recent work carried out in the wider context of the IST-ADRENALIN project, to facilitate formation and lifecycle management of networked enterprises. The projects focus is in designing and building an execution “kernel” for mobile agent applications written in Java, using the Aglets platform and integrating it with any existing ERP system. For integrating mobile agent infrastructures in operational ERP systems, we developed a generic model, which we call Mobile Agent Model, and which encompasses two parts concerning the Agent Logic and the Agent Proprietary Data, analysed in more detail in this paper. This model is branch independent and builds on the Adrenalin Company concept, where the fractal and the Information supply chain concepts are combined, introducing that every process, activity and resource can be defined with the “triangle” of executor – controller - co-ordinator tasks, supporting characteristics of self-similarity, self-organisation, self-optimisation and dynamic organisational behaviour.

1 INTRODUCTION

Emerging e-business patterns are forcing IT departments to build, deploy, and operate increasingly complex multi-tier “environments” that cover a wide thematic span including functionality expected to be found in various ERP-related applications. While operations staff will continue to be responsible for the selection and deployment of end-to-end ERP systems management tools, infrastructure developers will be responsible for building and deploying infrastructure that is "management ready" and can be quickly "plugged" into the existing management environment.

To achieve this, our aim in ADRENALIN is to provide support to the infrastructure developer for defining an architectural blueprint for the management interfaces that cover monitoring, controlling, deployment and configuration of the particular infrastructure related to a specific corporate ERP system and the related applications. This must define a common set of standards for the management APIs as well as the protocols for accessing them. It must also drill down into each of the infrastructure platform components (e.g. the RDBMS, the Web server, the TP monitors, etc.) to define what must be managed and how. The adopted approach simplifies the set-up of new application scenarios after a new component has been deployed.

The projects focus is in designing and building an execution “kernel” for mobile agent applications written in Java, using the Aglets platform and integrating it with any existing ERP system. This model is branch independent and builds on the fractal idea by supporting self-similarity, self-organization, self-optimisation and dynamic organizational behaviours (Martinetz et all, 2000).

2 THE ADRENALIN COMPANY

The ADRENALIN Company Concept (ACC), is in this respect treated as a general model (approach) to specify, design and realize in a conceptual way how to organize Information Supply Chains (ISC) and routes, and navigate across networked enterprise activities.

Information Supply Chain methodology (Walsh and Koumpis, 1998), is a tool for identifying the information requirements of an enterprise for the design of an information system or a network application, by describing the business flow inside and outside an enterprise. Business flow consists of information and workflow inside and outside the component. In ISC method we are investigating the information flow between the components and the business flow inside the component. Enterprise Information Supply Chains (Walsh and Koumpis, 2000), are important communication entities in Fractal Companies (Warnecke, 1983).

Accepting that every resource, activity and process operates depending on the competence, authority and mission, which is assigned to do, we individually describe and define every process, activity and resource by a “triangle” of three basic tasks. These tasks may be viewed as the primary assignments that characterise and give life to a resource, activity or process and eventually to a business operation (Peitgen et all, 1992). More specifically, each networked enterprise activity may be described in terms of an operational synthesis of the following three tasks, (Figure 1):

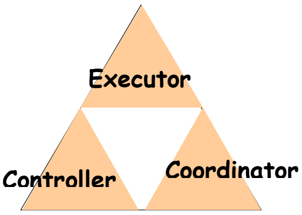

Figure 1: Triangle” of three basic tasks- assignments

- Executor, who is responsible for carrying out “basic” activities on the various resource types and has the ability to implement the assigned jobs;
- Controller, which is responsible for applying performance measurements (set by the co-ordinator) on the executor and represents the ability of the entity (resource, activity or process) to be kept in the boundaries that are defined by the co-ordinator;
- Co-ordinator, which is responsible for interfacing with other entities / systems / components and represents the ability of the entity to organise end regulate the assigned jobs.

3 THE PROGRAMMING MODEL

For integrating mobile agent (Cockayne and Zyda, 1997) infrastructures in operational ERP systems, we developed a generic model, which we call ADRENALIN Mobile Agent Model, and which encompasses two parts concerning the Agent Logic and the Agent Proprietary Data, also analysed in Figure 2.

- Agent Logic: Concerns the entire logic-related parts of the Agent, and is consisting of three parts for achieving optimisation of its functioning and adaptability to the particular execution environments. The aim of the "Agent Logic" is to facilitate all the needed functionality “nest” as well as to separate from the data so that different “Agent Platforms” can easier implement it.
- Agent Platform: It is the particular platform under which the execution of the Mobile Agents takes place. It may either be one of the existing mobile agent platforms, e.g. JavaSeals (as in our case of IBM Aglets), or one that may emerge in the market in the near future. In this way we ensure the capability to increase the utilization frontier for the developed infrastructure.
- Communication Lib: It is the connective node between the platform and the rest of the Agent. Eventual change of the platform would necessitate changes in the particular communication libraries, so that any new platform would be supported. At the physical level, it is a set of library modules, which are compiled into a pre-selected set of instructions into data and method invocations of the Agent.
- Logic: It is the underlying logic according to which the Agent will perform its scenarios. The logic component differs substantially in case the Agent is executed on the Server (Mobile Agent Server or MAS) or in case it is executed in a Mobile Agent Client (MAC). The "Logic" part is scripted in a way complementary to the libraries.
- Scenario: In our case it concerns the combined ADRENALIN scenario concerning combined work and information flow; it is loaded, executed and transferred dynamically from the MAS to the respective MACs and may be stored to any RDBMS, thus enabling connectivity to any existing (operational) legacy system.
- Object Data: They are the data, which can be alternatively Classes or instances of Objects, their respective Graphical User Interfaces (GUIs), or database variables that are transferred and interact with the particular scenario. "Object Data" are platform independent and communicate with the Agent by invoking selected components from the "Communication Lib".

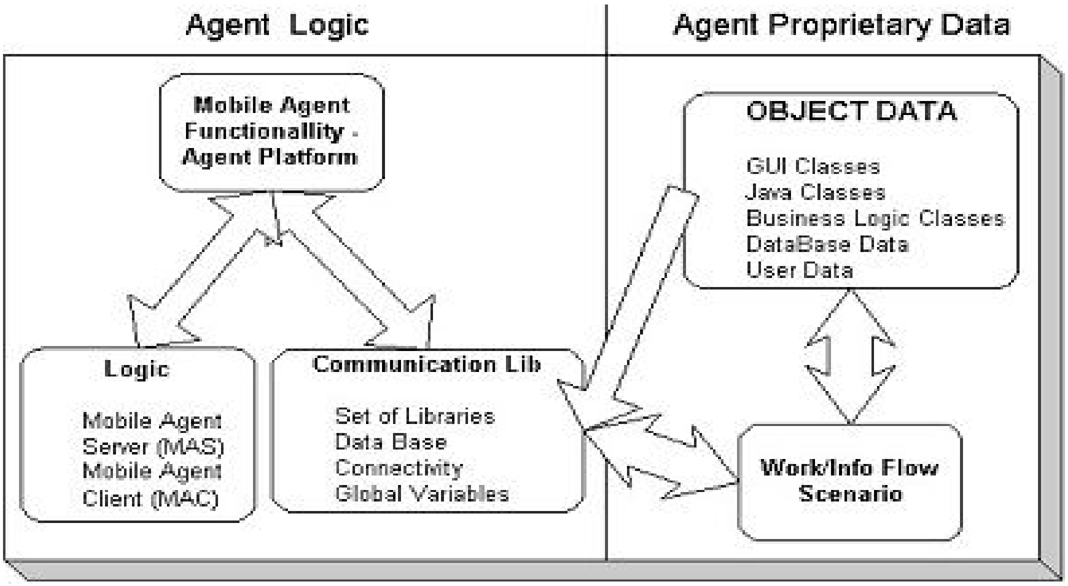

Figure 1: ADRENALIN Mobile Agent Model

4 USAGE SCENARIOS

As stated before, the Aglets library provides developers with a set of APIs to build and configure applications using Aglets technology. With one of these APIs, an application can have its own configured viewer, import a portion of server facilities, or create and launch an Aglet without server capabilities. There are several further elaborated scenarios for building an application by using Aglets technology as approached in ADRENALIN. While a server can host running Aglets, a client can create and control an Aglet remotely without any Aglet context. Info Chain users may want to use Aglets through their web browsers. In this case, applets can be used either as a context or a client, although it is not yet possible to dispatch an Aglet to a browser (Lange 2000).

4.1 Accessing Databases

A client creates an Aglet to gather information from several databases. The client provides the Aglet with an itinerary and sends it off to the first database server. The itinerary controls the routing of the Aglet and is able to handle exceptional conditions. For example, if a server is not responding, the Aglet will go on to other servers first and then return to that server later.

On each database server, the Aglet can be engaged in heavy data retrieval without suffering from network latency, since the database is accessed locally at a very high bandwidth. Furthermore, the possibly mobile client can be disconnected from the network while the information gathering takes place. The Aglet will return when the client is eventually reconnected. The database structure and server topology are not revealed to the network.

This typical scenario does not require any additional APIs. The application only consists solely of a set of Aglets built on top of the standard Aglet API. Such a configuration normally includes a stationary Aglet that has privileged access to local resources such as databases and files systems, and provides services for incoming Aglets. The classes used to define the stationary Aglet can be from either local disk or a given remote base. If it's loaded from the local disk, no security limitation are needed to become enforced by the security manager, the Aglet has the highest privileges. If it is loaded from a remote base, appropriate security enforcement will be applied in accordance with its identity.

With this configuration, an incoming Aglet normally gains access to services via message passing. Then, it can leave for the next server along with the result obtained in the server, or send it to the home server by remote message passing and die. Message passing is also under the control of the security manager and a receiver Aglet can also deny a request.

4.2 Web Site Monitor

A Web browser can dispatch an Aglet to a Web site. The Aglet can perform a local search among the documents and information on that site or it can monitor specific documents / information for updates. When a document / information entity is updated the Aglet can e.g. either send an email to its owner or return to its origin.

The client can search and monitor the Web even while it is disconnected, and monitoring of Web sites does not require a continuous network connection. For Web sites, it means that interested readers are automatically and instantaneously notified of important updates. Furthermore, a Web-crawling Aglet can move between Web sites to perform high-bandwidth local searching or index building.

This scenario is a typical way of adding the functionality of accepting and hosting Aglets into a "traditional" application. For example, in a simple info chain implementation accommodating functionality found in a generic groupware application, we may use Aglets to provide service and to use it, or to communicate with the other group members. It would be inadequate for such applications to just use the Aglet server as it is, even if the developer can implement its own viewer. The standard Aglets Server API allows developers to embed and bootstrap the Aglets server in their applications, and to configure the server's SecurityManager, Persistence, Viewer, and so on.

5 INTEGRATION WITH THE ERP SYSTEM

"Fundamental" requirements of the ADRENALIN Company for integrating with the ERP, are the following:

Enterprise Integration: In order to support rapid responsiveness, an enterprise is integrated both at the organisational and the IT levels with related management systems (e.g., purchasing, orders, design, production, control, resources, personnel, materials, quality, etc.) of its partners.

Distributed Organisation: For effective enterprise integration across distributed organisations, distributed systems will be needed so as to link for instance demand management directly to resource and capacity planning and scheduling of the corporate Extranet of Suppliers and Customers.

Heterogeneous Environments: Such systems will need to accommodate heterogeneous software and hardware in both their organisational and IT environments.

Interoperability: Heterogeneous information environments may use different technology platforms, programming languages, represent data with different representation languages and models, and operate in different organisational frameworks.

Open and dynamic Structure: It must be possible to dynamically integrate new subsystems into or remove existing subsystems from the system without stopping and reinitialising the working environment.

Scalability: Scalability means that additional resources can be incorporated into the organisational and IT infrastructure as required. This capability should be available at any working node of an Information Supply Chain and at any level within the nodes.

Fault Tolerance: The end system should be fault tolerant in terms of enabling recovery from failures at any level – lexical, syntactic or semantic - and minimise their impacts on the working environment and on the particular business process.

6 ACKNOWLEDGEMENT

Significant part of the reported work has been partly funded by the European Commission through IST Project ADRENALIN: Advanced Fractal Companies use Information Supply Chains (No. IST-1999-10181). The authors wish to acknowledge the Commission for support and the ADRENALIN project partners for their contribution during the development of various ideas and concepts presented in this paper.

REFERENCES

Martinetz, J., Markfort, D., Klostermeyer, A., and Brehmer, N., 2000. Advanced, Fractal Companies use Information Supply Chain. *E-business: Key issues, applications and technologies*. Ohmsha, (IOS Press).

Walsh, P., Koumpis, A., 1998. *Introducing the concept of Information Supply Chains: The Buddy Project*, Journal of Logistics Information Management, Vol. 11, No. 2.

Walsh P., Koumpis A., 2000. Analyzing efficiencies in networked enterprises using Information Supply Chains, in *e-Business and e-Work Conference, Madrid, Spain, 18– 20 October 2000*.

Warnecke, H.J., 1983. *The Fractal Company: A Revolution in the Corporate Culture*, Springer-Verlag, Germany.

Peitgen H.O, Jurgens H., Saupe D., 1992. *Chaos and Fractals: New frontiers of Science*, Springer Verlag, Germany.

Cockayne W. , Zyda M., 1997. *Mobile Agents*. Manning Publications.

Lange D. and Mitsuru O., 2000. *Programming and Deploying Java Mobile Agents with Aglets*. Addison