# Automated Engineering of e-Business Processes
# The RosettaNet Case Study

**Giacomo PICCINELLI, Anthony FINKELSTEIN**
**Dept. of Computer Science**
**University College London**
**Gower Street, London WC1E 6BT, UK**

**and**

**Eric STAMMERS**
**Hewlett-Packard E-Process Operation**
**Nine Mile Ride - Pinewood Park**
**Wokingham RG40 3LL, UK**

## ABSTRACT

For more than twenty years, process automation has been successfully used to improve efficiency within companies. The resources involved in the production of goods and services were predominantly internal, and such situation was reflected by process models and execution technologies. With business models now shifting towards a more cooperative approach, process automation needs to absorb the requirements coming from newly engineered business processes.

The efficiencies generated by process automation depend substantially on the stability of the process specification. The upfront investment in the engineering of a process is spread over the instances of the process actually executed. Reconciling this type of requirements with the dynamic nature of business-to-business (B2B) relationships is the main objective of the DySCo (Dynamic e-Service Composer) process automation platform developed at HP Labs. The process model in DySCo is based on multi-party orchestration, functional incompleteness, and dynamic service composition.

In this paper, we describe our experience using the DySCo platform for the modelling and implementation of the partner interaction processes (PIPs) proposed by the RosettaNet consortium.

**Keywords:** e-Business, Business Processes, RosettaNet

## 1.INTRODUCTION

Business processes [11] are among the key intangible assets of a company. The role of process automation systems is to capture the operational knowledge of a company and organise the activity of business resources accordingly.

The resources and objectives involved in a business process impose very specific requirements on the software systems supporting such processes [23, 24]. For example, business processes are first engineered by people with business expertise, and then mapped to formal specifications in the language of the process automation system. On the technology side, properties like reliability and availability are often pushed to the limits of dependability. Commonly referred to as workflow management systems or process management systems [5], process automation technologies are a fundamental component of almost any ERP solution (Enterprise Resource Planning).

Process automation has a long history in both the business and technical world, but attention has focused mainly on the management of internal processes. EDI (Electronic Data Interchange) solutions had only a limited impact on this tendency. Recent developments of information technology and cooperation-oriented business models are instead setting new trends for the engineering of business processes, and supporting technology [18]. The e-service model revolves around the automation of the interaction capabilities of a company [1, 4]. Evolving from the basic notion of company web site, the move is towards a complete automation of customer interaction; including aspects like product offer and delivery. E-marketplaces represent the new point where offer and demand meet, and electronic negotiation takes place.

DySCo (Dynamic Service Composer) is a process automation platform developed at HP Labs to explore and address automation requirements of business processes involving e-marketplaces and e-services (referred to as e-processes) [12, 16]. The companies involved in an e-process and the roles that they play depend on the outcome of a negotiation activity within e-marketplaces [14, 15]. Different companies can choose to cover a different set of roles in different instances of an e-process. Using DySCo, the process projections associated to a set of business roles are generated automatically. When each company executes the projection associated to the agreed roles, it can be formally proved that the semantics of the overall process is respected.

In this paper, we explore the interrelations between DySCo and the e-process framework proposed by the RosettaNet consortium [19]. One of the objectives for DySCo is to fully automate the engineering of multi-party e-process solutions in distributed environments.

RosettaNet processes represent a case study that reflects the requirements and the complexity of a real world application. All the partner interaction processes (PIPs) currently specified by RosettaNet were mapped into a library of DySCo cooperative steps. The library was then used for the development of a number of complete e-processes.

After an overview of both DySCo (Section 2) and RosettaNet (Section 3), we describe the way in which RosettaNet PIPs were implemented and used in DySCo (Section 4). An example of the complete lifecycle for an e-process based on PIPs is presented (Section 5). An analysis of related works is presented in Section 6, and in Section 7 we present some final considerations.

## 2.THE DYSCO FRAMEWORK

E-business processes have a number of special requirements, in addition to those of standard internal processes [2]. Resource management and availability are major points of differentiation [17]. Crossing company boundaries imposes strong limitations on the level of access to the resources involved in a process. In e-processes, resources are modelled as service modules (e-services). E-services are bound to specific interaction and behavioural specifications that the provider contractually commits to enforce.

From both a modelling and a management point of view, e-processes have a coarser granularity than traditional internal processes. For example, there can be a company offering a printing service instead of a printer. From a printer, an internal process can only expect sheets of paper. A specific sub-process is required to package the pages into a business report, and send it to a customer. From a printing service, the e-process can expect a document to be printed, packaged, and even sent directly to a specific recipient. Having a report printed and sent to a customer can be considered as an atomic action in an e-process. If a document is not properly reproduced by the printer, the internal process has to cope with a technical fault. If the same document is not properly reproduced by the company offering the printing service, the cooperative process has to cope with a breach of contract.

The modelling and enforcement of multi-party e-business processes is the focus of the DySCo (Dynamic Service Composer) framework. DySCo has been developed based on concepts like business-to-business cooperation, service aggregation, and dynamic re-configuration of business roles [16]. In the remaining part of this section, we first give an overview of the business model supported by DySCo. We then present the process model and the development platform currently part of the DySCo framework.

### Business model

The evolution of e-marketplaces is having a strong impact on the way business relationships are formed and maintained [20]. E-marketplaces aggregate supply and demand generated by a well-defined set of organisations, and enable negotiation among the parties. From the company perspective, the shift is from a static set of customers and suppliers to a set of e-marketplaces. From the relationship with an e-marketplace the company then derives relationships with a dynamically changing set of customers and suppliers.

When they first appeared, e-marketplaces were used mainly as an extra channel for selling excess capacity, or to compensate for the temporary stock problems of existing suppliers. Today, e-marketplaces represent an important component of the business architecture for a growing number of companies. Existing business processes have been reengineered in order to exploit the new dynamics of supply and demand that derive from the use of e-marketplaces. Companies are also experimenting with entirely new business models, and investing into the development of the business processes to support such models [18].

The processes targeted by the DySCo framework derive from a business scenario based on the dynamic nature of e-marketplaces. In particular, DySCo address the problem of having different companies playing different sets of roles in different instances of the same process. The objective is to reconcile the need for flexibility in business relationships with the need for stability and reusability of business processes.

### Process model

In order to understand the process model used in DySCo, it is first necessary to understand the resource model used in the framework. For the reasons discussed at the beginning of this section, the resources involved in an e-process derive from the capabilities made available by the participating companies. In DySCo, the concept of *Role* is used to represent a unit of business capability (unit of service) that will be required at execution time. The role abstracts from the operational characteristics of the entity that will actually play the role itself. An organization usually plays more than one role. To link the abstract concept of role within the process description with the execution environment of a process instance, the concept of role group is used. A *Role Group* captures the set of all the roles played by the same organization. Examples of roles are transport provider, quality tester, packaging provider, and customer. An example of role group could be buyer. If the buyer company takes care of transporting and quality testing the goods, the three roles in the role group could be customer, quality tester, and transport provider.

Inter-company cooperation is at the core of e-processes. The process model used in DySCo is based on three levels of abstraction: cooperative framework, cooperative process, and cooperative step. A *Cooperative Step* is an atomic unit of activity inside a cooperative process. It can involve a single role, or it can require the coordination of two or more of roles. The actions performed by the roles within the step involve data exchange between specific roles (pass), data sharing using a logical storage space available to all the roles in the step (share), and specific service-oriented activities (process). At the beginning of the step, a set of data from the process is made available to all the roles involved in the step itself (input data). A set of significant data is also kept at the end of the step (output data), and externalised at the process level. An example of a cooperative step is the transfer of the goods from the transport provider to the customer. A *Cooperative Process* is a process meant to orchestrate the execution of a specific aspect of the cooperation between a number of business partners. The elements of a process are mainly represented by cooperative steps required of specific set of roles. The process model and related process description language follow the standard reference defined by the WfMC (Workflow Management Coalition) [9]. A typical example of cooperative process is the payment of an

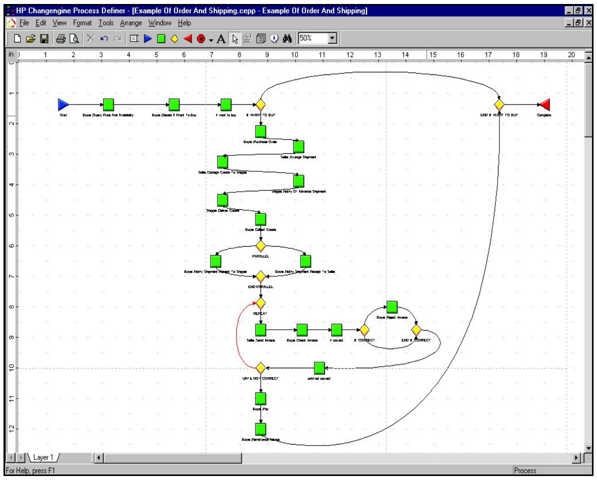


**Figure 2.a:** Process design

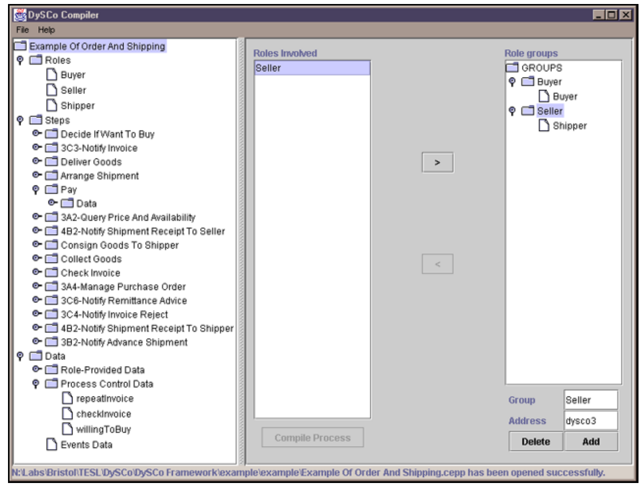


**Figure 2.b:** Projection generation

invoice, which could involve roles like customer, producer, bank, and archive provider. A *Cooperative Framework* represents a set of inter-communicating cooperative processes. Examples of cooperative frameworks are the customer interaction framework, the order management framework, and the production support framework. In terms of communication model, messages are used within cooperative processes and cooperative steps. Events are used for inter-process communication.

The last fundamental aspect of the process model in DySCo revolves around the concept of projections. The *Projection* of a cooperative framework onto a role group embodies a view on the framework limited to roles in the role group. A company should be concerned only about the part of a cooperative framework related to the role group it is contracted to fulfil. The content of a projection captures the activities related to a given role group, plus the coordination required with other role groups. Projections are used during the negotiation phase in order to express the level of commitment required to a company. Projections are also used for the execution of the business processes contained in the cooperative framework. Central to DySCo is the capability to automatically generate projections (16).

**Development platform**

The development and execution platform is an integral part of the DySCo framework. Main aspects in the lifecycle of a cooperative business process are the design, the projection, and the enactment. We first introduce individually the modules of the platform. We then present an end-to-end use example.

A requirement for the overall DySCo framework was to consider the mapping between e-business processes and internal processes. In terms of process automation platforms, the requirement was to investigate the gap between existing process management technology and that required to support e-processes. The approach we chose was to develop the DySCo platform on top of a standard commercial product for business process management. The product selected was HP Process Manager (HPPM) [8].

Standard process models have been designed mainly for internal processes. Still, the flexibility of the WfMC model and of the HPPM design environment (Figure 2.a) allowed a complete mapping of the DySCo cooperative model. The languages for the cooperative processes and the cooperative steps were engineered as a set of patterns to be applied to standard process design in HPPM. Patterns specify how to model elements like synchronisation points, exchange of information between roles, service requests, inter-process communication, and flow control. Processes are described using an XML-based notation called XpML [8]. A detailed description of languages and patterns used in DySCo is outside the scope of this paper [15]. In the following sections, there will be the opportunity to see a number of examples of use of process patterns.

The generation of the projections for the processes contained in a cooperative framework is managed by a compilation module specially developed for the DySCo platform. The projection module was designed mainly for programmatic use, but a graphical interface for manual use is also available (Figure 2.b). First, role groups are defined and roles assigned to the groups. Projections can then be generated automatically. From a business perspective, projections describe the expected interaction behaviour for the companies accepting specific business roles. In order for the set of projections to reproduce the semantics of the original cooperative process, implicit synchronisation is required in addition to the explicit synchronisation deriving from the process description. An example of explicit synchronisation is the submission of an order form. The supplier has to wait for the order form to arrive before fulfilling a customer request. Implicit synchronisation derives instead from the partial view that each organisation has on the overall process. Among other functions, the projection module automatically introduces all the required synchronisation elements in the definition of the projections.

As an example of process decomposition, let us assume that in a branch of the cooperative process there are three cooperative steps (S1, S2, S3) supposed to happen in sequence. Company C1 could play roles involved in S1 and S3, but not in S2. Company C2 could play roles involved in

S2, but neither in S1 nor S3. The projection for C1 should not contain activities deriving from S2, and the projection for C2 should not contain activities deriving from S1 or S3. Nevertheless, in order to preserve the semantics the global process, C2 cannot start any of the activities in S2 before C1 and the other companies involved in S1 have completed all the activities in S1. Similarly, C1 should not start any of the activities in S3 until C2 and all the companies involved in S2 have completed the step.

The enactment environment in the DySCo platform is composed by the HPPM process engine, and by a number of modules specifically developed to handle activities like synchronisation, data exchange, and user interaction. Each company participating in a cooperative process has a private instance of the execution environment where it can deploy and run the projection it agrees to play. From HPPM we derive process execution capabilities, as well as a complete management and monitoring infrastructure. The process projections combine two distinct levels of abstraction. Some of the steps enforce business logic. An example is the step that requires an order form to be provided. These types of requests are routed to the user interface. Some steps derive instead from the need to handle synchronisation and the flow control. The activity involved in these steps is fulfilled by specific modules in the platform, and it doesn't propagate to the user.

To summarise, the process definition module (Figure 2.a) is normally used at the beginning of the lifecycle of a cooperative process. The process definition is then passed on to the projection module (Figure 2.b) for partitioning. Different grouping configurations can be necessary for the roles involved in the process. The projection generator automatically generates the set of projections associated to each configuration. Once deployed on the various instances of the execution platform, projections coordinate among each other to drive the enactment of the global e-process.

## 3.THE ROSETTANET STANDARD

RosettaNet [19] is a consortium of more than 400 of the world's leading electronic components, information technology and semiconductor manufacturing companies. The declared mission of the consortium is to drive collaborative development and rapid deployment of Internet-based business standards creating a common language that enables open e-business processes. The intent of RosettaNet is to offer standards for data dictionaries, implementation frameworks, business message schemas, and process specifications.

The cooperation model proposed by RosettaNet has the Internet and XML as a foundation. These first two layers offer basic data exchange capabilities. Moving up the abstraction layers, there are dictionaries and frameworks. These two layers capture the words and the grammar for business communication. Business dialogues are modelled as simple processes called PIPs (partner interaction processes). PIPs can then be organised into more complex cooperative business processes. The work described in this paper focuses mainly on RosettaNet PIPs and cooperative business processes.

The specification of each PIP encompasses three different levels of abstraction. These levels are also referred to as views. The specification includes a business operational view, a functional service view, and an implementation framework view. The business operational view (BOV) captures the semantics of business data entities and the flow of exchange between roles as they perform business activities. The functional service view (FSV) specifies the network component services and agents, and the interactions necessary to execute a PIP. In RosettaNet terms, a service is an entity that implements protocols that include a service layer, transaction and action layer. A description for the various layers that goes beyond the intuitive semantics deriving from the names can be found in the literature [19]. The service has an identity that can be registered in directories and used for component communication in a distributed computer system. An agent is an entity that implements protocols that include the action layer and the agent layer. There is no service layer or transaction layer. The implementation framework view (IFV) specifies elements like network protocol, message formats, and other communications requirements.

As an example, in the PIP for querying technical information (PIP 2.A.5) the business operational view first specifies two roles: the product information user and the product information distributor. The product information user uses technical information for updating enterprise systems and online promotion systems such as electronic catalogue systems. The product information distributor distributes technical information to both fulfil technical information subscriptions and to respond to queries for technical information. The BOV then describes the types of messages exchanged between the two roles, in terms of content and exchange flow. The exchange logic for the messages involved in the request can be found in [19]. Messages are specified using XML-based structures. The functional service view is also described in [19]. A meaningful discussion of the implementation framework view cannot be captured in the space of this section. The full description of the IFV can be found in the RosettaNet standard [19].

The alignment between the cooperation model proposed in RosettaNet and the one proposed in DySCo emerges in particular in the distinction between internal and external processes. In both models, cooperative processes coexist with internal process and they complement each other. With reference to the cooperative stack presented at the beginning of this section, the RosettaNet standardisation effort has currently reached the level of PIPs. A complete description and categorisation of RosettaNet PIPs can be found in [19].

In the following sections, we present a mapping of the currently finalised RosettaNet PIPs into the DySCo framework.

## 4.THE PIP LIBRARY

The partner interaction processes specified in the RosettaNet standard capture atomic conversations between two business roles. Examples of PIPs are queries for price or availability of products, and notification exchanges related to shipment, invoicing, and payment. In order to port the PIPs into the

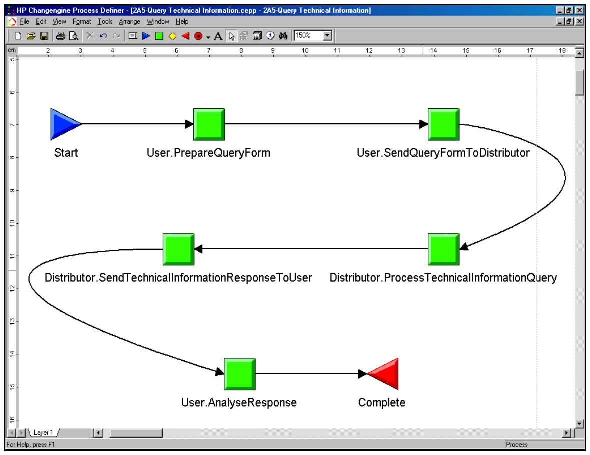


**Figure 3:** PIP 2.A.5 Query Technical Information

DySCo framework, the approach we took was to capture the business operational view for the PIPs into DySCo cooperative steps. A library of cooperative steps was developed containing all the currently specified PIPs.

From a structural perspective, all PIPs adhere to one of three main patterns. The first pattern is a single action activity. The role that starts the PIP produces a document, and passes it on to a second role. The second role just receives the document and performs an action. An example of PIP of this type is the PIP 2.A.8 for the distribution of product stock-keeping units. The second pattern involves two actions. One role starts the PIP producing a document and passing it on to a second role. The second role receives the document, performs an action, produces a response document, and sends it back to the first role. The first role receives the answer and performs a final action that concludes the PIP. An example of PIP following this pattern is the PIP 2.A.5 for queries related to technical information. The last pattern involves a choice from the role that starts the PIP. The choice has a binary outcome, and both branches of the process reflect one of the two previously presented patters. An example of this type of interaction is the PIP 3.A.4 for the management of the submission of a purchase order. The encoding of the PIPs into DySCo cooperative steps reflects the patterns of their business operational view.

As an example, the cooperative step associated to the PIP 2.A.5 for querying technical information is presented in (Figure 3). In the first node, the role in charge of starting the PIP (the User) prepares the document that has to be sent to the second role involved (the Distributor). In the second node, the user actually sends the document to the Distributor. In the third node, the Distributor receives the document and performs an internal action. The internal action indicated here represents the point of contact with the internal processes of the Distributor. The Distributor has to produce a document containing the response to the query, or some form of notification. In the fourth node, the response is sent to the User. In the fifth node, the user receives the document and performs an internal action. The analysis done by the User of the response received from the Distributor ends the cooperative step.

The level of complexity involved in the mapping of a PIP into a cooperative step is linearly dependent on the complexity of the business operational view of the PIP. The complexity of both the functional service view and the implementation framework view are hidden from the designer, and are automatically handled at the lower levels of the DySCo platform. The projection generator will introduce all the required synchronisations, and the instructions for the platform components in charge of the data exchange for a cooperative process.

## 5.A PROCESS IMPLEMENTATION EXAMPLE

The purpose of this section is to illustrate the use of the DySCo platform for the engineering of an e-business process involving RosettaNet PIPs. The emphasis is on the support provided by the platform for the design and deployment of the solution, rather then on the business logic of the process.

The process considered derives from a typical customer interaction framework, and focuses on order management. Three business roles are involved: Buyer, Seller, and Shipper. The Buyer requests to the Seller information about goods, and then decides if it wants to place an order. If the Buyer places the order, the Seller has to arrange for the shipment. At this point, the Shipper is involved. The Shipper has to collect the goods from the Seller and deliver them to the Buyer. When the Buyer receives the goods, it sends information about the conditions of the goods received to both the Seller and the Shipper. Then the payment phase starts. The Seller sends an invoice to the Buyer. If the Buyer agrees with the invoice, it pays. If it doesn't agree with the invoice, a dispute resolution cycle is started. Payment concludes the process. A flavour for the structural complexity of the process can be derived from (Figure 2.a). Eight PIPs appear as cooperative steps, including PIP 3.A.2 (Query Price And Availability), PIP 4.B.2 (Notify Shipment Receipt), and PIP 3.C.6 (Notify Remittance Advice).

The design of the e-process is done only considering the business roles involved and the interaction expected among them. The roles defined in the PIPs are instantiated with the roles used in the overall process. The fact that the shipment will be done by the company acting as a buyer, by the one acting as seller, or by a third party is transparent to the designer. Once the process is defined, the business roles can be aggregated into groups. Different group configurations are possible depending on the companies involved in an actual instance of the process. The information on the role groups is passed to the projection module (Figure 2.b), which automatically generates the processes required to the cooperating companies. As for the other parts of a cooperative process, a projection contains only the part of a PIP related to the roles in a role group. Optimisations are possible for the case in which the two roles involved in a PIP are covered by the same company.

## 6.RELATED WORKS

An overview of the general requirements for e-business frameworks can be found in [22]. While processes seem to be a common denominator, different approaches have been followed in terms of the logical and physical distribution of a cooperative process.

For example, in Interworkflow [7] the cooperative process is directly mapped on to internal processes. A two-layer model for internal and cooperative processes like the one of RosettaNet [19] and DySCo [16] can be found in Cross Flow [10, 21]. A more centralised approach is instead used in COSMOS [6], Aurora [13], and in [11].

A standardisation initiative similar to the one of RosettaNet initiative is ebXML [3].

## 7.CONCLUSIONS

The flourishing of standardisation initiatives like RosettaNet and ebXML is an indicator of the interest that both the industry and the research community are showing for open e-business processes. Open standards are the foundation of business-to-business interaction, and there is a need to make them easily available to the designers of business solutions. Beyond the specification of standards, the concept of availability extends to the resources that process designers require for enforcing a standard.

In this paper, we first presented a conceptual framework for e-business processes (DySCo), together with a reference implementation. We also describe the way in which the partner interaction processes (PIPs) specified by RosettaNet are mapped into the framework in the form of a specific process library. A use case illustrates the application of DySCo in the development and enactment of a multi-party purchase process.

## 8.REFERENCES


[1] Blodget H., and McCabe E. *"The B2B market maker book"* Merrill Lynch & Co., 2000.

[2] Chen Q., Hsu M., and Dayal U. "Peer-to-peer collaborative Internet business servers" *HP Labs Technical Report*, HPL-2001-14, 2001.

[3] ebXML Initiative. Reference Web Site 2001, http://www.ebxml.org

[4] Gipson M., Runett R., Wood L., and Clawson P. *"The electronic marketplace 2003: strategies for connecting buyers and sellers"* Simba Information Inc., 1999.

[5] Gruhn V. "Business process modeling and workflow management" *International Journal of Cooperative Information Systems*, Vol.4 N.2-3, 1995, pp. 145-164.

[6] Griffel F., Boger M., Weinreich H., Lamersdorf W., Merz M. "Electronic contracting with COSMOS - How to establish, negotiate and execute electronic contracts on the Internet" *In 2nd Int. Enterprise Distributed Object Computing Workshop* (EDOC '98), 1998.

[7] Hiramatsu K., Okada K, and Hayami H. "Interworkflow system: coordination of each workflow system among multiple organizations" *Proc. 3rd IFCIS International Conference on Cooperative Information Systems*, New York City, USA, 1998 pp. 354-363.

[8] Hewlett-Packard. *HP Process Manager*, Reference Web Site, 2001, http://www.hp.com/go/e-process

[9] Holligsworth, D. "*The workflow reference model"* Workflow Management Coalition, WfMC, TC00-1003, 1994.

[10] Klingemann J, Wäsch J., Aberer K. "Adaptive outsourcing in cross-organizational workflows" *Proc. 11th Int. Conference on Advanced Information Systems Engineering (CAiSE'99)*, Heidelberg, Germany, 1999.

[11] Koulopoulos T.M.*"The workflow imperative. Building real world business solutions"* John Wiley & Sons, 1995.

[12] Marton A., Piccinelli G. and Turfin C. "Service provision and composition in virtual business communities" *Proc. 18th IEEE Int. Symposium on Reliable Distributed Systems, Int. Workshop on Electronic Commerce*. Lausanne, Switzerland, 1999.

[13] Marazakis M., Papadakis D., Nikolaou C. "*The Aurora architecture for developing network-centric applications by dynamic composition of services"* Technical Report TR97-0213, Institute of Computer Science, FORTH, 1997.

[14] Milosevic, Z. and Bond, A. "Electronic commerce on the Internet: what is still missing?" *Proc.5th Conf. of the Internet Society*, 1995.

[15] Piccinelli G., Preist C., and Bartolini C. "E-service composition: supporting dynamic definition of process-oriented negotiation parameters" *Proc. IEEE 2nd e-Negotiations Workshop, part of 12th International Conference on Database and Expert Systems Applications (DEXA)*, Munich, Germany, 2001.

[16] Piccinelli G. and Mokrushin L. "Dynamic e-service composition in DySCo" *Proc. Int. Workshop on Distributed Dynamic Multiservice Architecture*, part of the 21st IEEE Int. Conference on Distributed Computing Systems (ICDCS-21), Phoenix, Arizona, USA, 2001.

[17] Podorozhny R.M., Lerner B.S., Osterweil L.J. "Modeling Resources for Activity Coordination and Scheduling", *Springer Verlag Lecture Notes in Computer Science* N. 1594, 1999, pp. 307-322.

[18] Reilly B. and Block J. *"Next-generation e-commerce processes and systems"*, Electronic Commerce Strategies Report, GartnerGroup, 1997.

[19] RosettaNet Consortium. Reference Web Site, 2001, http://www.RosettaNet.org.

[20] Timmers, P. "*Electronic commerce - Strategies and models for business-to-business trading"* Addison Wesley, 1999.

[21] Sachs M., Dan A., Nguyen T., Kearney R., Shaikh H., and Dias D. "Executable trading-partner agreements in electronic commerce" *IBM Technical Report*, 2000.

[22] Shim S., Pendyala V., Sundaram M., Gao J., "Business-to-business e-commerce frameworks" *Computer*, Vol.33 N.10, IEEE Computer Society, 2000, pp. 40-47.

[23] Sommerville I., and Sawyer P. *"Requirements engineering: a good practice guide"* John Wiley & Sons, 1997.

[24] Wald E., and Stammers E. "Out of the alligator pool: a service-oriented approach to development" *EAI Journal*, March 2001, pp. 26-30.