Towards Adaptive Cooperative Process Management Based on e-Services

Enzo Colombo *, Valeria De Antonellis †, Chiara Francelanci *, Massimo Mecella †,
Michele Melchiori †, Barbara Pernici *, Pierluigi Plebani *

*Politecnico di Milano
†Università di Roma “La Sapienza”
‡Università di Brescia

contact author: Barbara Pernici
Dipartimento di Elettronica ed Informazione - Politecnico di Milano
Via Ponzio 34/5 - 20133 Milano (Italy)
e-mail: pernici@elet.polimi.it
Abstract

Cooperative process management based on e-Services requires support for design, retrieval, and composition of e-Services. In the VISPO project the authors have studied the characteristics of an e-Service-based approach in virtual districts. The paper illustrates the VISPO approach to design coordination in a virtual district, proposed mechanism for coordination and substitution of e-Services in a dynamically evolving environment, and an ontology-based approach to retrieval of e-Services.

I. INTRODUCTION

In virtual enterprises and virtual organizations [1] different companies pool together their services to offer more complex, added-value products and services, made easily accessible through network technologies and Internet.

Many approaches have been proposed for supporting such business paradigms; by using an approach based on inter-organization business process coordination [2], the cooperation among different organizations is obtained by sharing, integrating and coordinating services across networks; such services, commonly referred to as e-Services or Web-Services [3], [4], are exported by different organizations as well-defined functionalities that allow users and applications to access and perform tasks offered by back-end business applications.

Within the Italian VISPO (Virtual-district Internet-based Service PlatfOrm) project, we aim to create a flexible environment to support the design of service-based cooperative processes and the dynamic coordination of e-Services in a district-based scenario, by considering not only the design of complex e-Services starting from simple ones, but also the dynamic substitution of failed or modified e-Services.

In the business reality, a district is a consortium of independent small and medium enterprises, which operate in an integrated and organic way, often on the basis of personal and informal relationships, to exploit business opportunities. The goal of the project is to consider the application of information technology to such realities, in order to (semi-)automatically create business relationships (not necessarily based on geographical aggregation) and thus obtaining virtual districts.

In the proposed approach, cooperating enterprises follow predefined process specifications when interacting; the structure of the process and the modalities of interaction are designed according to the nature of interactions in the district, ranging from completely hierarchical structures to open markets [5]. Then processes are defined as the composition of abstract services. Only at execution time, given an abstract service specification, a suitable concrete service is retrieved and invoked. The VISPO architecture supports the execution of dynamically evolving cooperative processes based on e-Services: in particular, mechanisms for run-time substitution of e-Services with compatible ones and for adapting e-Services are being developed.

In the paper we discuss in Section 2 the characteristics of virtual districts and the design of an e-Service-based infrastructure for them. In Section 3, the VISPO architecture is discussed, and in Section 4 the concepts of similarity for retrieval and substitutions of e-Services are illustrated.

II. VIRTUAL DISTRICTS AND e-SERVICES

Outsourcing represents a form of cooperation based on inter-organizational coordination mechanisms. The literature highlights how outsourcing is beneficial to cooperating organizations when it leverages process
synergies and is accompanied by greater organizational flexibility [6], [7]. Industrial districts are a typical example of cooperative environments that are effectively based on pervasive outsourcing. An industrial district can be defined as a geographical aggregation of decisionally and financially independent organizations cooperating along a common value chain. Within a district, organizations are usually small or medium sized (SMEs) and cooperation is ruled by a multitude of outsourcing relationships along the value chain. The large number of inter-organizational relationships and the overall frequency of transactions suggests high information intensity, which is associated with significant technological opportunities.

As an example, Figure 1 represents a sofa realization process, which is typical in one of the districts in which VISPO results are being experimented. The sofa design, fabric cutting, frame creation, and sewing and stapling phases represent the strategic activities that mark the quality of the sofa. Typically, these phases are performed internally by the district-leader enterprise. Conversely, material procurement, cleaning, warehousing, and delivering activities, indicated with grey filling in the figure, can be performed by e-Services provided by other organizations.

However, information technology has been poorly exploited to support inter-organizational cooperation within districts [8]. The current emphasis of numerous research programs on the need for creating virtual districts indicates that the design of inter-organizational information systems within industrial districts remains a major open issue of the IS literature [9]. From an organizational perspective, virtual districts raise critical research issues related to the complexity of coordination mechanisms. Virtual districts involve the implementation of a variety of coordination mechanisms, ranging from market to hierarchical coordination [10], [5]. Different coordination mechanisms involve specific control procedures and inter-organizational information dependencies, which, in turn, raise technology issues to be accurately specified, designed and implemented.

A primary technological factor preventing automation in virtual districts is the heterogeneity of information systems. Different organizations within the same district have different technology standards for hardware, network, data and software. The adoption of a common technology platform to favour cooperation is obviously
challenging, especially due to the economic barriers raised by legacy systems. Recently, e-Services have emerged as a conceptual paradigm to enable cooperation among heterogeneous information systems [4]. E-Services wrap organizations information systems and provide a standard interface that enables cooperation with other organizations information systems. They can be viewed as the public representation of the part of an information system that is meant to be used by other organizations belonging to the same district. E-Services also embed additional features that allow the representation of inter-organizational workflows as a set of cooperating e-Services.

With respect to traditional workflows, cooperating e-Services can dynamically adapt to changing environmental conditions, such as the availability of new e-Services of higher quality or the failure of an e-Service and consequent need for replacement. Therefore, e-Services are defined as software artifacts designed to cooperate automatically. This involves their automatic synchronization, composition and overall coordination towards a common goal.

In VISPO, a cooperative process together with its governance mechanisms is specified in the following steps:

1) **Specification of Cooperation Requirements.** Our notation is based on the concept of goal, task, resource and decision redefined with the purpose of modeling cross-organizational workflows [8]. A goal is defined as the objective of delegating. A task is referred to as a structured sequence of activities as a means to produce an added value transformation of inputs into outputs. In our reference scenario examples of task are material procurement, sewing and stapling and so on. Resources are what an organization needs to make its business, they can be both material or immaterial and range from an organization’s products and services to raw material and information. We are interested in representing cross-organizational interactions; consequently a resource is simply referred to as an information needed to execute a task and it is therefore associated with the concept of dependency. Typically, when an organization has more than one business partner, coordination problems among partners during task execution can occur due to the lack of parallelism among the activities performed by each organization. Coordination mechanisms are therefore needed to manage dependencies among the activities executed by each of the business partners. Traditionally, this type of dependencies are classified as value dependency and commit dependency [11]. Besides, a distinctive feature of our approach is that concept of decision is tied to exception management. Decision represents a shift in decision power when exceptions occur.

2) **Design of Inter-organization Business Transactions.** A first step to analyze the structure of inter-organization business transactions is to understand the most appropriate mechanism of cooperation (market, hierarchy or combined forms) in a given context; such a mechanism is selected by evaluating product complexity (including entry barriers), asset specificity (monopoly or not, differentiation on costs and quality of service), the cardinality of the cooperation paradigm, and the necessity of performing control activities due to critical functional characteristics of the product [12], [10]. The corresponding transaction patterns are selected for each e-Service.

In Figure 2 the different cooperative pattern of business transactions defined in VISPO for market and hierarchical structures are defined. Each pattern ensures that the e-Service is performed correctly in a given context, through appropriate preparation and quality control phases. A fundamental difference between
market and hierarchical transactions is that the latter implement coordination according to a mediated conversation among the counterparts, where the mediator is the principal organization in the hierarchy. This interaction paradigm is more complex than the request-response paradigm characterizing market control. However, a market transaction is functionally richer than a hierarchical transaction, even if the quantity of information exchanged during hierarchical coordination and control is larger than in market transactions.

Four types of market transactions are defined, each including matchmaking (whereby potential business partners become aware of each other’s existence) and execution phases, and possibly including negotiation and post-settlement phases. Three types of organization hierarchy transactions are defined, each including execution and post-settlement phases, and possibly including the management of commit and value dependencies between controlled organizations.

3) **Design of the Coordination and Control Mechanisms.** According to organizational theory [5], cooperation is defined as a pattern of control and coordination mechanisms ruling interactions among organizations; control is defined as the activities performed by cooperating partners to ensure that goals are achieved and coordination is defined as the sequence of messages that allow cooperating partners to execute their operating activities. Cooperation is enacted through the execution of business transactions. The patterns govern the execution of transactions by (i) identifying the message protocols among

---

### Table: Notation

<table>
<thead>
<tr>
<th>Notation:</th>
<th>Beginning of Business Transaction</th>
<th>End of Business Transaction</th>
<th>Logic flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>n: phase, x: actor</td>
<td>{μ}</td>
<td>{π}</td>
<td>{μ},{π}</td>
</tr>
</tbody>
</table>

### Diagrams

**Fig. 2. Patterns of cooperation between organizations**

- **(a) Market Transactions**: (1: Matchmaking, 2: Negotiation, 3: Execution, 4: Post-settlement)
- **(b) Organizational hierarchy transactions**: (1: Manage Commit Dependency, 2: Execution [2a: execution business process, 2b: Manage Value Dependency], 3: Post-settlement)
cooperating actors, that is the behavior during a cooperation interaction, and (ii) specifying the type of data exchanged during the interaction. At this step, designers formalize control and coordination interactions associated with cooperative business transactions by selecting the corresponding control and/or coordination patterns. Designers should:

- Instantiate the control and/or coordination patterns associated with each cooperative business transaction defining a) the number of instances per pattern (i.e., three control interaction between the supplier and the customer) and b) identifying the content of the exchanged message according to the specification provided in Step 1 and 2 (i.e. involved tasks, goals to be fulfilled, etc).
- Formalize how to compose more than one control and/or coordination instance to create complex cooperating sessions among organizations. The resulting pattern should be classified and stored to be reused in the future.
- Specialize, if necessary, control and coordination patterns to deal with particular interactions (hybrids belonging to the continuum between market and hierarchy). A session is specialized by isolating a part of its behavior, and by using this part as new behavior. An exchanged message is specialized by adding new data types. The resulting pattern should be classified and stored to be reused in the future.
- Formalize temporal constraints on control and coordination interaction to prevent deadlocks.

4) **Orchestration and e-Service Design.** We assume that each activity described in a process schema such as the one illustrated in Figure 1 is specified as an abstract e-Service using a common language specification; such a language allows specifying:

- **interface:** provided operations and exchanged information using the WSDL [13] language;
- **behavior:** (partial) order of operation execution specified using WSTL [14] and pre- and post-conditions;
- **quality of service (QoS):** e-Service features, like availability, performance level, cost, and so on. A language for the specification of these characteristics is under development.

Given a process specification in terms of abstract services, a number of concrete services is selected from a set of available e-Services in a VISPO registry. Concrete e-Services are selected according to their compatibility with the abstract e-Services, as illustrated in Section V, and they may be substituted with other e-Services if they become unavailable. We define the *compatibility class* as the set of all e-Services that can be substituted with each other. All the identified services are related to an abstract description that represents all the members of the class: this abstract description is called the *abstract service* whereas the members are the *concrete e-Services*.

To be correctly invocated, an appropriate interface has to be provided for the selected e-Services to wrap them - on the invocation side - providing two different types of functionalities:

- **adaptation wrapper:** selected e-Services may be similar but not fully identical to the corresponding abstract e-Service specification, according to the rules described in Section V. In this case the invocator has, for instance, to map parameter names or type, or to provide wrappers to collect missing
information;

- **governance wrapper**: the core e-Service need to be enriched, as discussed before, by a set of properties and methods which enable cooperation. For instance, considering negotiation, negotiation methods enable the buyer to enquire the available sellers to obtain more detailed information on the core e-Service’s features. The negotiation wrapper adds to the core e-Service properties tied to service contents, quality of service and price. With the end of the negotiation phase and consequently with the choice of the seller that best fits requirements, the execution phase can be initiated.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Governance Paradigm</th>
<th>Transaction Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Material Supply</td>
<td>Accessories: Market</td>
<td>Basic form of Market Cooperation[see Figure 2.a.c]</td>
<td>Negotiation to purchase accessories. Low product complexity.</td>
</tr>
</tbody>
</table>
| Row Material Supply | Leather and Fabrics: Hierarchy | Complete form of Hierarchical Cooperation[see Figure 2.b.c] | - Value dependencies with suppliers of expanded polyurethane and wood stalks.  
- Control activities managed by the principal organization. |
| Production    | Expanded polyurethane: Hierarchy | Complete form of Hierarchical Cooperation[see Figure 2.b.c] | - Value dependencies with leather and fabrics suppliers.  
- Control activities managed by the principal organization. |
| Production    | Cut and sewing: Hierarchy    | Complete form of Hierarchical Cooperation[Figure 2.b.b] | - Commit dependencies with business partners producing expanded polyurethane and wood stalks.  
- Control activities managed by the principal organization. |
| Assembly + Quality Control | Managed in house (no governance) | - | - |
| Delivery      | Market                       | Complete form of Hierarchical Cooperation[Figure 2.a.d] | - Negotiation to select the best express courier.  
- Control activities managed by the principal organization (delivery tracking). |

**TABLE I**

**COOPERATION BETWEEN THE PRINCIPAL ORGANIZATION AND ITS BUSINESS PARTNERS**

An illustration of the model is discussed on the value chain of the sofa-production industrial district of Matera, Italy. Table I reports the main phases of the district value chain, the governance paradigm that is most frequently adopted, the transaction schema that best fits the relationship between buyer and seller and a comment on the
control and coordination schemas that should be instantiated.

Table I also highlights that the procurement of different goods, respectively accessories and leather, can require different governance paradigms, respectively market and hierarchy, to be correctly managed. In fact, leather and fabrics in the examined context are critical for the production of the final good, requiring a particular know-how to be produced and delivered, and therefore their purchase is likely to follow a hierarchical organization, while the procurement of accessories is easier to outsource according to a market paradigm. The procurement of accessories is modelled by a transaction which embeds matchmaking, negotiation and execution (see Figure 2.b.c). The procurement of leather is implemented by the hierarchical transaction of Figure 2.a.c. The instantiation of the e-Service corresponding to the supply of leather occurs when commit dependencies are resolved. The following additional governance wrapping on the core e-Service are needed:

- the control wrapping, which provides the methods to verify the output of tasks (i.e. controls the process of production of the leather and the respect of milestones);
- the coordination wrapping, which enables a mediated conversation between e-Services executed by different business partners to manage value and commit dependencies (i.e. the leather supplier needs information on the quality of expanded polyurethane to start the production of leather);
- the post-settlement wrapping, which verifies the overall output when the execution e-Service terminates (i.e. quality control and lead time control).

III. The VISPO Architecture

One of the main focus of the VISPO project is to define and develop a platform which supports the composition of dynamically evolving cooperative processes, defined according to the approach outlined in the previous section.

The Figure 3 represents the result of this work. By using this architecture, the retrieval, composition, and invocation of the e-Services provided by the enterprises belonging to the district is performed. Moreover a coordination mechanisms is realized in order to control the execution flow.

The main modules of the architecture are:

- **Registry**: it maintains the list of available e-Services using an augmented version of UDDI Registry [15] where, besides the tModels, the e-Services are classified using the semantic information provided by the service descriptor.
- **Cooperative Process Specification Repository**: it stores the cooperative process specifications in terms of execution flow of the activities, which correspond to the abstract services.
- **Compatible Service Provider**: given a particular cooperative process, for each abstract service involved, this module identifies the correspondent compatibility class and selects a subset of its members that better match the abstract service requirements.
- **Compatibility Module**: using both terms and service ontologies, this module supports the Compatible Service Provider in order to create and maintain the compatibility classes which group e-Services in sets where the members can be substituted with each other.
• **Invocation Module**: it selects and invokes the best concrete service inside the compatibility class of the abstract service that the Orchestration Engine decided to execute.

• **Orchestration Engine Module**: it monitors the Invocation Module activity in order to identify the active concrete service and to handle possible critical situations raising the appropriate exception. In VISPO, the orchestration engine is distributed, i.e., orchestration schemas are enacted by distributed orchestration engines, deployed by the different cooperating organizations.

The relationships between these modules are also represented by the UML Sequence Diagram in Figure 4. Using this approach the Orchestration Engine communicates with the e-Services only asynchronously. The Invocation Module(s), selects concrete services, and transforms all the requests of the Orchestration Engine in the respective method invocations and provides wrapping mechanisms. Note that the standard **create** and **destroy** methods are used to instantiate a service and to provide to the Orchestration Engine a reference to the real activity performer.

**IV. CoORDINATION MANAGEMENT Model**

The orchestration of e-Services is specified through an orchestration schema, which specifies interactions among them. Since multiple orchestration modules are cooperating to enact the cooperative process it is necessary to specify both the control and message flow and global process coordination. In VISPO, a specific orchestration engine is then in charge of interpreting orchestration schemas and to effectively coordinate e-Service instances, possibly dynamically substituting them. In VISPO, orchestration schemas are enacted by distributed orchestration engines, deployed by the different cooperating organizations; global control may be moved all along the enactment from one organization to another, although we assume that it is assigned only to one organization at a time.

e-Services instances, before being engaged in a specific enactment, need to be associated to such an
orchestration instance; such operation is referred to as linking, and freeing the e-Service instance at the end of the conversation is referred to as unlinking. While an e-Service instance is linked to an orchestration instance, it cannot be engaged in any conversation with other orchestration instances. Linking can be executed on the basis of specific e-Service properties, and the orchestration schema allows to specify conditions on which to execute the linking. Linking can be performed either at the beginning of the enactment of the orchestration instance (static allocation mode), i.e., all resources needed to execute the orchestration instance are retrieved in one shot at the beginning, or as the enactment proceeds, by linking e-Services as they are needed (dynamic allocation mode).

Therefore the orchestration of e-Services which support a cooperative process consists of (i) linking instances of the e-Services to be used in a specific enactment of the process, (ii) conversating with them, that is receiving messages sent from an e-Service and sending them to other one, and (iii) assigning and moving the task of the overall coordination to organizations.
A Coloured Petri Net-based model [16] is adopted to model orchestration, thus providing designers of the overall cooperative applications with appropriate tools for correctly assembling different e-Services; for example, deadlock freeness of the overall process and reachability of the final configuration of the involved e-Services can be verified by analyzing the configuration graph of the net.

In the net, specific places are used to represent message exchanges and states of both the e-Services and the overall process, and transitions are used to represent evolution of the process and to specify linking conditions and assignment of the overall control task.

An orchestration instance which refers to an e-Service $S$, can use an instance $V$ of an e-Service $V$ instead of an instance $S$ of $S$; this is referred to as substitution of e-Services. Substitution, as detailed in the following sections, is triggered when catching specific exceptions.

V. THE ONTOLOGY-BASED e-SERVICE COMPATIBILITY APPROACH

In order to support either the matchmaking phase and the e-Service substitution, the retrieval functionalities described in the following, provided by the compatible service provider, can be used.

The approach is based on the concept of compatibility class defined as a set of e-Services that provide the same functionalities (e.g., procurement services rather than delivering services). The generation of such a compatibility class requires a deep knowledge about the features of e-Services in both syntactic and semantic aspects. In particular, the semantic aspects capture the purpose of the service and the context in which the service operates. The syntactic aspects, on the contrary, identify how and in which order the e-Service methods can be invoked.

A compatibility class is generated in association to a given e-Service description representing required functionalities in the execution of a cooperative process. The given e-Service description is called abstract service whereas the compatibility class members are called concrete services.

Moreover the mapping information, stating the rules which have to be satisfied when an e-Service is substituted with another, is defined. Following this information the concrete service can be wrapped in according to the difference with respect to the abstract service. The wrapper can be also enriched with other functionalities in order to support the selected governance mechanism.

Our approach for the construction and use of compatibility classes consists of the following steps [17].

Publication. The service provider publishes e-Service specifications in a private and augmented UDDI Registry: the VISPO Registry. To perform a semantic analysis of services on the basis of their own interface, in our approach a service descriptor [18] is generated starting from a WSDL description. In particular, a service descriptor provides a summary, structured representation of the features of an e-Service that are relevant for compatibility assessment.

Classification. Under the supervision of a domain expert, a semantic analysis is performed on e-Service specifications stored in the VISPO Registry. The e-Services are classified in compatibility class according to behavior-based and similarity-based [19] analysis.
The behavior-based analysis relies on exact/partial match relationships defined on the basis of [20].

The result of the classification is the skeleton architecture of a Domain Service Ontology that organizes e-Service specifications according to the following semantic relationships:

- if an exact match is established between \( S_i \) and \( S_j \) then a \( S_i \text{equivalent-to} S_j \) relationship is added to the ontology.
- if a partial match is established such that \( S_i \) subsumes \( S_j \), then a \( S_j \text{is-a} S_i \) relationship is added to the ontology.
- if \( S_i \) and \( S_j \) are in the same cluster according to the similarity-based classification, and there is no partial or exact match between them, a \( S_i \text{similar-to} S_j \) relationship is added to the ontology.

To increase ontology interoperability for Semantic Web, a DAML-S-based [21] representation of e-Service descriptions and semantic relationships in the Domain Service Ontology can be given.

Semantic relationships in the domain service ontology allow the definition of compatibility classes. In fact, given an abstract service definition, the services semantically related to it in the Domain Service Ontology belong to the same compatibility class.

Retrieval. Once defined the compatibility class, the retrieval phase uses the different created relationships among the e-Services to select a subset of it according to the degree of equivalence requested by the service requestor. Moreover the mapping information, stating the rules which have to be satisfied when an e-Service is substituted with another, have to be defined.

During a matchmaking phase, where the result has to be a set of e-Services equivalent to a given e-Service, the retrieval phase derives from the Domain Service Ontology only the services linked to the given e-Service according to the relationship equivalent-to.

On the contrary, if during the process execution an e-Service fails and it is mandatory to recover immediately the process, the retrieval phase will have to select all the concrete services belonging to the compatibility class of the failed e-Service.

In any case, once obtained a set of compatible services, the choice depends on quality factors negotiation requests.

VI. CONCLUDING REMARKS

In this paper we have presented the VISPO approach to adaptive cooperative process management based on e-Services. The approach, on the basis of a process schema agreed between cooperating partners, allows the dynamic selection and substitution of e-Services for performing activities. Particular attention has been given to different modalities of interaction in a virtual district, ranging from pure hierarchy to market organization. Each modality requires different coordination and control mechanisms, in addition to providing services to execute the activities. Cooperation is performed in a distributed environment: we assume that each cooperating organization is autonomous from the other ones, and in particular that no centralized workflow management system is provided and that global control on a process is performed through a responsibility delegation mechanism. Combined with the possibility of dynamically substituting e-Services as the process evolves and in different
instantiations of the process, the VISPO approach goes in the direction of providing peer-to-peer mechanisms for process coordination in an inter-organizational environment. Such mechanisms are also based on providing functionalities for retrieval and classification of e-Services, based on an ontology-based approach.

Further work is needed on the design of coordination mechanisms and modalities of interaction, on mechanisms to wrap e-Services to adapt them to the specific needs of a process, and on e-Service classification and composition mechanisms. Future work will also study extensions of the VISPO approach to support mobile information systems, with moving autonomous units with coordination and control functionalities.

ACKNOWLEDGMENTS

This work has been partially supported by MIUR through the “Fondo Strategico 2000” Project VISPO (Virtual-district Internet-based Service PlatForm) (http://cube-si.elet.polimi.it/vispo/index.htm) and the “FIRB 2001” Project MAIS (Multi-channel Adaptive Information Systems) (http://black.elet.polimi.it/mais).

Thanks also to IBM and Prof. Frank Leymann for providing the IBM Web Service platform on which the prototype is based on.

REFERENCES


