Appears in *Proc. of ISPE Concurrent Engineering (CE)'02, Cranfield University, UK, 27-31 July, 2002*

e-Alliance: a software infrastructure for concurrent inter-organisational alliances*

Ilham Alloui[♯], Jean Marc Andreoli [†], Olivier Boissier [‡], Mihnea Bratu [‡], Stefania Castellani [†], Karim Megzari [♯]

[†] *CT/XRCE 6 chemin de Maupertuis, 38240 Meylan, France*

[‡] *MAS/SIMMO/ENSM.SE 158 Cours Fauriel, 42023 Saint-Etienne Cedex, France*

[♯] *LLP/CESALP 5, Chemin de Bellevue, BP 806, 74016 - Annecy Cedex, France*

* *e-alliance is a Thematic project funded by the Région Rhône Alpes 2000-2003, France.*

In this paper we introduce e-Alliance, a software infrastructure we are defining for supporting negotiation activities in concurrent inter-organisational alliances. e-Alliance main intent is to preserve autonomy of organisations within an alliance while enabling concurrency of their activities, flexibility of their negotiations and dynamism/evolution of their environment. The IT infrastructure we propose combines different technologies, such as software engineering techniques, middleware-level coordination facilities and multi-agent systems support. We present our approach in the context offered by a sample scenario where business-to-business interactions hold among printshops grouped into an alliance to better answer customers' demands.

1 INTRODUCTION

Workflow management tools have proved successful to organize in a structured way the collaboration between well defined processes with a relatively static, well established overall goal (eg. the realization of the objectives of a virtual enterprise, or the handling of commercial relations between a major manufacturer and its network of sub-contractors). At the other end of the spectrum, tools for computer supported collaborative work aim at very dynamic and essentially unstructured (or loosely structured) collaborations between processes. In this paper, we are interested in the intermediate situations between these two extremes, where fully autonomous organisations execute and coordinate their own processes internally, but rely on some long-lived infrastructure in which to expand these processes if needed, in order to collaborate, with minimal work to setup such collaborations. We call this infrastructure a virtual alliance (or *e-alliance*). It is more structured than a shared dataspace for collaborative work, but less structured than a workflow assigning precise roles to participant organisations and tightly scheduling their interactions. An e-alliance provides the computational environment for opportunistic collaborations, which may occur asynchronously, and may follow very different patterns, but are nevertheless constrained by general rules of behavior agreed upon by the participants in the alliance. Tools for supporting such alliances typically have the following characteristics:

- *Open*: an organisation may dynamically join or leave an alliance;
- *Non intrusive*: each partner in an alliance makes autonomous decisions to progress in its collaborations;
- *Negotiation-oriented*: negotiation is considered as the fundamental mechanism for the partners of an alliance to collaborate.

In order to illustrate our purpose, we consider here the case of an *alliance of autonomous printshops*, which are autonomous organisations, fully responsible for their budget and for the planning and scheduling of their print jobs and resources. There is no central broker (or principal) which would gather the external customers'requests, process and dispatch them to individual printshops (sub-contractors) based on their state revealed to the broker through strict procedures. Instead, each printshop locally manages its contracts, customer portfolio, machines, schedules etc. On the other hand, joining an alliance may help a printshop accomplish some of its customers'print requests that it cannot or does not want to satisfy alone.

The IT infrastructure, called "e-Alliance", proposed here to support e-alliances combines different technologies, such as software engineering techniques, middleware-level coordination facilities and multi-agent system support.

Sec. 2 illustrates a typical example of the kind of scenario the e-Alliance infrastructure aims at providing support for. Sec. 3 describes the issues we address in providing support to e-alliances, the requirements guiding the design of the e-Alliance infrastructure and the main goals. Sec. 4 provides an overview of the e-Alliance approach and Sec. 5 concludes the paper.

2 SCENARIOS

We are interested in studying negotiations interactions and patterns among distributed autonomous organisations, grouped into alliances to improve their own business ability. We consider here alliances of organisations offering similar and/or complementary competencies and capabilities, competing but also collaborating with each other to improve their own ability to accomplish for example customers' requests. As already mentioned, the collaborations within an alliance can be partially formalized and automated as workflows, but they cannot be satisfactorily modeled by simple activity diagrams with rigid dependencies defining only synchronizations and ordering between activities. A lot of flexibility is needed for example to support negotiations activities.

For exploring issues on how to provide the support for collaborations inside this kind of alliances, we consider a scenario where distributed autonomous printshops, offering similar and/or complementary print competencies and capabilities, are grouped into an alliance. The scenario has been defined building upon the description of a variety of print work processes and it describes the activities in a printshop, including its interactions with the customers and possibly with other printshops. We do not consider the interactions for the creation of an alliance that is out of the scope of this paper.

2.1 *Adhesion to an alliance*

A printshop manager is interested in joining a printshops alliance including a number of printshops and a committee. Then, (s)he contacts the alliance (e.g. through his(er) web site) to ask informations about the objectives of the alliance, who are the potential future partners, what are the rules, etc. The manager provided with the necessary information can then formally ask for joining the alliance as a new member (e.g downloading a "membership request form" called in the sequel adhesion contract). (S)he will then provide information on the resources, competencies and preferences of his(er) printshop. The alliance's committee may then accept or refuse the printshop as a new partner. This decision may be based on the complementarity of services for instance. In case of acceptance, the new member should respect the rules of the alliance and his(er) adhesion contract.

2.2 *Out(In)sourcing of jobs within the alliance*

Printshop managers receive print requests from customers and schedule their portfolio of print jobs (simply called jobs in the sequel). A printshop manager may wish to negotiate with a number of selected allied printshops the outsourcing of a (part of a) job (s)he cannot or do not wish to perform locally. The manager will then select, among the printshops making insourcing offers, those providing for example the best cost/color-quality performance ratio. Then a printshop may act sometimes as an "outsourcing" entity, submitting job requests to other printshops in the alliance, and sometimes as an “insourcing” entity, accepting such requests.

Printshops’s customers can be customers outside the alliance or other printshops partners of the alliance. When a print request reaches the printshop with given parameters (e.g. deadline), a first estimation is established. The manager analyses the job description to understand how it can be accomplished, taking into account the current job schedule, the availability of the resources, and trying to optimise the global cost (e.g. using ad-hoc scheduling tools can be used here like shown in (Castellani et al. 1998)). Based on the results of this evaluation, the manager decides either to reject or to accept the print request. In the former case, the negative decision is communicated to the requester. In the latter case, the job has to be allocated. If the current schedule of the printshop allows inclusion of the new job, the manager may decide to perform it locally. However, it may be possible that the job cannot be locally performed (at least not as a block), given the requirements, the printshop resource availability and technical capabilities. For example, if the request includes a color print and the printshop has only black and white printers, then at least the part requiring a color printer should be outsourced. Moreover, even if the execution of the job is consistent with the printshop schedule and equipment, the manager might still decide to outsource (part of) the job, for example, in order to save some of the available resources for a job currently under negotiation with a major customer.

If the manager decides that (part of) the job has to be performed remotely, (s)he will start a negotiation with the partners printshops. The manager can select the partners of the negotiation among the members of the alliance according to their declared capabilities and the knowledge (s)he has about them. The outcome of a negotiation can be “success” (the job was fully outsourced), “failure” (no outsourcing agreement could be reached) or “partial” (only part of the job could be outsourced). An example of negotiation scheme relies on an “invitation to tender” (as described in (Andreoli et al. 2000)). The manager decides if and how to split the job into slots and notifies the other printshops in the alliance about the outsourcing requests for the different slots. The manager collects quotations from partner printshops, evaluates them and chooses a solution. The outsourced job (or slots) is (are) then sent to the selected insourcer(s). If no “good” solution is found, the manager may accept a sub-optimal offer anyway (and possibly face delays), or re-allocate local resources in order to perform the job locally, or revise the splitting of the job. Also, the manager of an insourcing printshop may need to re-negotiate with the outsourcing printshop the commitment for a job, e.g. for changing a deadline.

2.3 *Contract management in the alliance*

In case the negotiation process ends in a “success” state, a contract is settled between the outsourcing printshop and the insourcing ones. The contract is a complex object which is the basis of trust in this coordination mechanism. It defines an inter-organisational workflow enacting the business process fulfilling the different outsourced and insourced jobs throughout one or several plans (sequential execution, intermelt execution, parallel execution, cyclic production ...) Moreover, it contains a set of specific “business” rules (e.g. penalties) expressing obligation relations between the participants. They participate to the definition of the way this workflow should be coordinated. The manager of the outsourcing printshop then monitors and supervises the execution of the business process, taking care of a successful execution of it.

In case of failure of a partner, the manager will have on one hand to manage the obligation relations, e.g. oblige the partner to finish his(er) job, set penalties, etc. and on the other hand to modify the business process, e.g. by imposing increasing constraints on the processing, or by renegotiating parts of the job that have not been realized.

2.4 *Management of the alliance*

An alliance is a dynamic entity as new printshops may join or leave it. Members may need to leave for many reasons, for instance, when they change their activity or when they do no longer want to collaborate with the partners of the alliance. In case of departure from the alliance, the leaving partner may either notify the alliance’s committee or all the partners. It also may leave without giving any information. In any case, the departure of a partner may result in problematic situations, in particular when the partner is involved in several ongoing jobs. The departure of a partner from the alliance will have an important impact on ongoing contracts in particular when this partner is an insourcer of an important amount of jobs. At least three situations are then possible: (1) the partner commits himself/herself to continue processing the current jobs; (2) the partner takes the initiative to delegate the remaining jobs to partners (s)he knows they are able to do them; (3) the partner leaves the alliance and remaining jobs are re-negotiated (shorter deadlines, much more resources, etc.) between the remaining partners.

Next section discusses the scenario described above and outlines some issues and requirements for the e-Alliance IT.

3 E-ALLIANCE REQUIREMENTS AND GOALS

The printshops alliance sample we have presented in the previous section is a typical example of the alliances the e-Alliance IT will support. We will use it to express in a first time the requirements and in a second time, the objectives of the e-Alliance IT.

3.1 *Requirements*

As we saw, alliances are characterized by a mix of co-operation and competition. Indeed the printshops which build up an alliance, are in a normal context in competition with each other. But under given conditions they may decide to cooperate in order to better accomplish the customers' demands and thus to be more competitive.

The members of the alliance are distributed autonomous printshops fully responsible for their budget and for the planning and scheduling of their print jobs and resources. Each printshop locally manages its contracts, printers, customer portfolio, schedules etc. Interactions involved in such a context are very general "business-to-business" interactions, that can be found in other alliance of organisations, whatever their domain.

Each partner in the alliance needs at least to:

- optimise the usage of its resources (people and/or machines);
- coordinate several concurrent activities at once;
- monitor activities to be able to react to unforeseen events and dynamically solve problems if any (e.g. problems with a printer);
- capture and log information on transactions to be able for example to bill a customer or for statistics purposes.

Furthermore, in the case of the printshops offering several document production services, the jobs they perform may include: (1) assembling an electronic version of the document, that in turn may require some transformations, e.g. format conversion from several source formats; (2) printing and binding documents according to given specifications, e.g. print quality; (3) distributing printed documents to final recipients.

Obviously, a lot of coordination and multiple communications among the partners are required to reach an agreement on how and when to achieve a customer's request and then to carry out the contract and to monitor its execution. Moreover, in such an alliance changes may happen at several levels: a printshop may enter/leave, an agreement is not reached to answer a client's request, a contract is revoked for a given reason, resources are not available, etc.

Therefore, an IT infrastructure for such kind of alliances should enable: (**i**) *autonomy* in decisions and dates of partners; (**ii**) *coordination* of concurrent activities and resources consumed in a inter-enterprises collaboration; (**iii**) capacity to *adapt* in a dynamic and changing environment in order both to handle unexpected events and to cope with new requirements such as involving a new partner in the alliance or managing consequences of the departure of a member, especially if involved in ongoing contracts.

The kind of alliances we are interested in can be found in several application domains: virtual enterprises, e-business, and e-commerce (in particular B2B). Several approaches have been proposed in research to support alliances in the mentioned application domains (Dignum and Sierra 2001)(Dignum and Cortes 2001). European projects (e.g. MEMO[1] and PRODNET-II[2]) and groups of interest (e.g. the AgentLink excellence network SIG dedicated to the electronic commerce[3]) are interested in developing environments offering new opportunities for inter-organisational coordination and intermediation. Institutions (e.g. IMS[4], (Hurwitz 1998)) are also working on these topics and several proposals are already available on the market (see for example ROSET-

[1] http://www.abnamro.com/memo/index.html

[2] http://www.uninova.pt/p̃rodnet/

[3] http://www.iiia.csic.es/AMEC/

[4] http://www.ims.org/

TANET[5], COMMERCENET[6] and CIIMPLEX[7]).

However, existing approaches to support alliances focus mainly on the management of orders among partners. They generally provide no support to negotiate contracts and if they do, it is always through market places that do not preserve confidentiality nor manage highly dynamic information such as job plans and schedules. Furthermore, no support is provided to manage alliances themselves, i.e. the global knowledge on partners and the ongoing contracts/negotiations is difficult to access.

3.2 *Goals*

The main goal of the e-Alliance infrastructure is to provide a software support for inter-organisational alliances that enables:

1. the management of an *alliance's* life-cycle, through services for information publishing, partners authentication, monitoring of the interactions, or services for joining and leaving the alliance;

2. collaborative activities among the alliance partners, through services that allows the partners to negotiate, execute and monitor contracts.

These two aspects are crucial to meet most of the requirements of open, autonomous (possibly heterogeneous) alliances. The e-Alliance infrastructure should flexibly support negotiations and collaborative executions initiated by printshops willing to outsource some of their jobs (as a whole or in parts). In particular, it should support the negotiation activities in the alliance taking into account the autonomy requirements of the printshops without statically attaching each negotiation participant a role according to a strict protocol. Also, the mechanisms offered to support such collaborations should be generic enough to adapt to any business-to-business context. Moreover, the e-Alliance infrastructure should help the organisations grouped into an alliance to augment their efficiency and ability to react to unforeseen situations, thus improving their market competitiveness.

These issues have not been completely explored until now although a lot of work has been done on other aspects like how to support an organisation when searching for potential partners of a new alliance or how to define payment mechanisms. We focus instead on the following aspects:

- using multi-agent systems models and techniques to represent decentralized organisations;

- modeling the coordination of different concurrent interactions;

- formalizing negotiations (agents representation, conversation protocol, strategies);

- deploying and maintaining an alliance during its life-cycle;

- creating and administrating job contracts.

Next section describes how the e-Alliance approach supports the kind of distributed organisations alliances we target taking into account the discussed issues and requirements.

4 E-ALLIANCE APPROACH

For each manager of a printshop belonging to the alliance, the e-Alliance infrastructure proposes a multilevel architecture in order to provide different services to assist him(er) in the collaborative concurrent activities described in the previous sections. From the goals stated in the previous section, several aspects must be taken into consideration:

- *Modeling and deployment of an alliance lifecycle.* This includes the definition of mechanisms supporting the evolution of an alliance and, for example, in case of a change in the alliance configuration, supporting the impact of changes on the active contracts partners can be involved in. The infrastructure should provide support for dynamic changes in the alliance configuration and contracts.

- Identification and modeling of the information that the alliance has the right to demand or to provide to a partner. *Confidentiality* and *dynamicity* of the information are aspects to be taken into account as well, especially if the alliance allows partners to dynamically join/leave the alliance.

[5] http://www.rosettanet.org/

[6] http://inf2.pira.co.uk/top031.htm

[7] http://ciimplex.org

- *Definition of the interaction processes.* First, negotiation should be *flexible*, *decentralized* and *reconfigurable* according to the evolutions of the alliance. It also should allow to split/group jobs, to outsource/insource jobs, to choose a group of favorite partners for a negotiation. Second, the final contracts has to specify the behavior and responsibilities of the partners and the hierarchy of the roles in every particular collaboration. All this information on these processes may be provided by the alliance to its members, e.g. when searching partners for a negotiation, the printshop could be informed both on the dependencies with the other partners, given the previous and current collaborations, and their descriptions as members of the alliance.

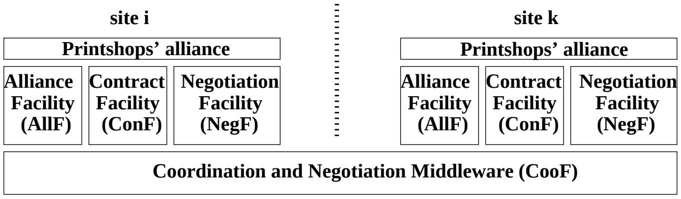

Figure 1: e-Alliance software infrastructure

Given these different characteristics, the e-Alliance infrastructure is organized in three layers as shown in Fig. 1. The three layers are defined as follows (from the more generic to the more specialized):

- a first layer is defined by a coordination and negotiation middleware (*CooF*);

- a second layer contains the alliance facility (*AllF*) the contract facility (*ConF*) and the negotiation facility (*NegF*). The *AllF* facility is dedicated to the management of the alliance lifecycle, while the *NegF* and the *ConF* facilities allow to manage the outsourcing and insourcing of jobs in the alliance;

- the third layer is a specialization of these facilities to the application domain, for example to the printshops alliance case.

It is worth noting that, in order to enable a decentralized negotiation and to maintain the independence and autonomy of the participants in freely choosing their strategies and partners, each printshop is equipped with a *NegF*, a *ConF* and an *AllF* that execute concurrently while interacting whenever necessary.

In this paper, we focus on the two first layers which contain the mechanisms and the data necessary to realize the functionalities of the scenario. Next sections describe these layers more into details.

4.1 *Alliance facility*

The *AllF* facility supervises the activities of the *NegF* and *ConF* facilities in order to check whether the rules of the alliance are respected or not. It also gathers information in order to build a global history of the system. When an event in the evolution of the alliance has an impact on the current negotiations and existing contracts (e.g. departure of a printshop), it interacts with the concerned facility in order to maintain the global coherence of the system. Its aim is to manage the global life-cycle of an alliance from its creation to its deletion. Several steps are concerned: creation of the alliance, subscription of a new member, modification of the adhesion contract and of the preferences expressed by a member in relation to the jobs that (s)he committed to outsource/insource in the alliance, modification of the global rules of the alliance, departure of a member.

The goal of the *AllF* is to model and enact dynamic and evolvable alliances. This addresses two major issues: (**i**) what kinds of software architectures cope at once with autonomy, openness and evolution requirements of alliances; (**ii**) what processes to put in place in order to specify, enact, deploy and evolve such alliances. In particular processes have to be defined to express the way the alliance life-cycle is managed, i.e. new partner registry, partner departure, alliance's history recording and management, etc.

At the level of the e-Alliance IT, these properties are provided through a software environment that offers the users means for dynamically adding/retracting/replacing software components, that is, without interrupting the system execution. *AllF* underlying language is a process architecture description language (PADL)(Alloui and Oquendo 2001) used to model the alliance's lifecycle. The resulted software environment is process-centered and allows the *AllF* to communicate with both *NegF* and *ConF* facilities. Communication is needed in both ways: from *AllF* to *NegF*/*ConF*, for instance to provide the *ConF* with the rules to apply to a given contract; from *NegF*/*ConF* to *AllF*, for instance to record in the history a new contract or a new negotiation.

Such open and architecture-based process support has already proven its relevance for decentralised, concurrent and heterogeneous environments within the framework of PIE (Process Instance Evolution) ESPRIT IV 34840 LTR Project (Alloui et al. 2000). The objective whithin e-Alliance is to combine such technology with both middleware-level coordination facilities and multi-agent system support in order to meet alliances' requirements we address.

4.2 *Negotiation facility*

The *NegF* facility aims at managing all the negotiations in which its own printshop is involved (e.g. as initiator or participant) with different partners of the alliance. It assists its printshop manager at a global level (negotiations with different participants over *different jobs*) and at a specific-level (negotiation over the *same job* with different participants) by coordinating itself with the *NegF* of the other partners by means of the coordination and negotiation middleware, *CooF* (described in next section).

Each negotiation is organized around three main steps: (**1**) initialization, by defining the attributes (called issues in the sequel) which characterize the job that will be under negotiation; (**2**) refinement, throughout exchanges with members of the alliance, (**3**) closing of the negotiation, by adopting and committing to jobs leading to the activation of the *ConF* facility. During this process, a *negotiation object* is manipulated. This data structure represents the different informations related to the negotiation under consideration.

The *initialization* of the negotiation process aims at defining the constraints over the issues under negotiation and the strategy of negotiation (e.g. time/resource/locally directed negotiation as in (Faratin 2000)) in the negotiation object. A first selection of possible partners can be made using history on passed negotiation, available locally or provided by *AllF*. The pre-selection of potential partners uses the *agents' modeling* approach proposed in (Vercouter 2000). This approach allows to model and manage the representation the participants can have on each other in a decentralized way. Thus, it differs from many existing work relying on broker, facilitator or yellow pages services. From this base, we can deepen the way in which an agent can exploit its representation of the others via the construction of networks of dependence relations according to the dependence theory of Castelfranchi implemented in (Sichman et al. 1994). The *negotiation* phase refines the object of negotiation throughout exchanges of proposals and counterproposals with the other participants in order to satisfy the constraints of both sides (outsourcer and insourcers). This phase relies on clearly specified speech acts (Carron et al. 1999), organized along interaction protocols. These interaction protocols define the elements and the functional rules of the negotiation process at the *NegF*'s level. In particular, they specify the *a priori* defined sequences of speech acts, the possible role and actions for a negotiation participant, the information that can be or must be exchanged in the process and how this information is represented - a common ontology for all the member of the alliance. We are working on the definition of the dynamic composition of these interaction protocols according to some global strategies of negotiations. The *adoption* phase commits over the job and activates the *ConF* facility. Locally, the resources involved in the job description are reserved, if a contract is signed.

This facility offers support for multi-issues, multi-participants (different partners can be involved) and multiple-cycles (cycles of proposals and counter-proposals over the same set of attributes) negotiations. These characteristics make the facility flexible enough for being used in different domains.

4.3 *Contract facility*

The *ConF* facility of a printshop manages the execution of the contracts with other partners of the alliance the printshop is involved in (as responsible or contractant). The management of a contract starts just after the adoption of a negotiation object in the negotiation process and lasts until its complete execution by the different participants. It is organized around the following main steps: (1) creation of the contract; (2) execution and supervision of its execution; (3) closing of the contract after its successful execution. In case of problems encountered during the execution, a modification step will take place in this cycle in order to repair or cancel the execution.

During the *creation*, the *ConF* of the principal contractant defines a contract from the negotiation object constructed during the negotiation by its *NegF* and the

NegF of the other negotiation participants. Besides the business process that is defined in order to fulfill the job concerned by the object of negotiation, a contract expresses a normative structure forcing the participants to behave in a certain way. Models of organisational structures in Multi-Agent Systems (Hannoun et al. 2000) allow to express the responsibilities of the *ConF* of each contract participant as *roles* (who ought do what) (s)he can play and the interactions with the other *ConF* as *links* (who has the power on whom) connecting these roles. Penalties are attached to these different obligations in order to penalize the participants which would be incapable of fulfilling the task under his(er) responsibility. The *execution* of the contract consists in the distributed execution of an inter-organisational workflow (corresponding to the business process expressed in the contract) between the different participants. It is supported by the coordination and negotiation middleware *CooF* which has facilities for the execution of distributed workflows (see next section). Different events may stop the execution of the contract implying in certain cases the modification of the contract itself. These events may be communicated by the *AllF* as a consequence of a change in the alliance itself (e.g. departure of a member which is a participant to the contract or modification of the adhesion contract). They can also be generated by the evolution of dynamic properties of each of the participants making them unable to fulfill the contract (e.g. change in the planning, failure of a machine, ...). The *ConF* will interact with the *NegF* of its printshop in case of necessity of launching a new negotiation. It will also interact with the *AllF* to make it aware of the different penalties that are set on the participants.

4.4 *Coordination and negotiation middleware*

The *CooF*, the coordination and negotiation middleware, is the support to the different processes that are provided by the facilities in the second layer. We are defining the *CooF* as an extension of the CLF middleware (Andreoli et al. 1999) aiming at enriching its negotiation support capabilities.

As a middleware toolkit, CLF has intentionally been designed to be independent of any application domain. Its building block is a very generic notion of resource, the theoretical foundation of which has been given in Linear Logic. The CLF notion of resource is extremely polyvalent, as it may denote rather tangible entities such as a file or a print job, as well as more virtual entities such as a task or a decision. A CLF component is therefore more than a simple black-box able to perform service invocations identified with method calls, as is traditional in object-oriented approaches to middleware (from Corba to the most recent Web services (Vaughan-Nichols 2002)). Instead, a CLF component is viewed as a resources manager able to accomplish requests for generic operations on these resources. An interaction with such components is then attached to one (or more) of its resources.

The interaction among the components is expressed through protocol primitives that define the generic operations on the resources that all the components offer. These primitives can be grouped into two main classes. The primitives of the first class allow to perform monotonous operations, such as discovery, selection and addition of resources. The primitives of the second class allow to consistently perform destructive resource manipulations. The first category of primitives is directly inspired by the events notification protocols of distributed systems ("publish-subscribe"), while the second category is inspired by the transactional protocols ("two-phase-commit").

This approach of systematically considering components as resources managers allows the coordination of these components by means of high-level scripts that completely hide the communication protocol and directly express the desired resources manipulations. Specific CLF components, called *coordinators*, manipulate scripts as resources. When a script resource is inserted in a coordinator, the abstract resource manipulations that it specifies are translated into invocations of the protocol on different components to be coordinated. Thus, the coordinators can be considered as generic clients of the middleware platform and the client side of a CLF application can be expressed as a set of scripts. These scripts can be dynamically inserted into multiple (distributed) coordinators during the execution of the application, and manipulated as any other resource by other scripts.

In order to better support the coordination of the various components of e-alliances, we are working on an extension of the CLF protocol to enable more powerful multi-party, multi-directional, multi-attribute negotiation interactions among the components (Andreoli and Castellani 2001), allowing flexible refinement of the negotiation terms.

On the server side, CLF offers a rich library of classes for building servers able to answer the protocol. A number of typical servers (databases, file systems, etc.) have currently been encapsulated as ready to use CLF components. Some basic components of this middleware have been used in several prototype applications for supporting “workflow” (Andreoli et al. 1996).

Building upon this experience, we are currently elaborating CLF-based support tools for the execution and monitoring of contracts in e-Alliance, transparently offering transactional and notification capabilities. The resources involved here represent the state of progress of the contracts execution, as well as the different actions performed by the contractors. The possibility of manipulating the scripts themselves as resources, for example by means of other scripts, offers the flexibility required for manipulating the contracts.

5 CONCLUSION

We have introduced e-Alliance, a software infrastructure we are defining for supporting negotiation activities in concurrent inter-organisational alliances. We have described the problems addressed by our work and how we approach them. A sample scenario where distributed printshops federate into an alliance has been used for illustration purposes.

This work is at the crosspoint of different domains of research in computer science: multi-agent systems for all the aspects related to negotiation and contracts, software engineering techniques for all the aspects related to alliance management, and finally middleware-level coordination facilities for all the aspects dealing with coordination and negotiation as support to the other technologies. This project enforces the idea that concurrent activities should be supported by such a composition of technologies if we want to develop applications that are open, non-intrusive and negotiation oriented.

Ongoing work includes refining the provided facilities in terms of functionalities, and interfaces. In particular prototypes to demonstrate the feasibility of our approach are under design. On the other hand, requirement elicitation and technology validation are undertaken in collaboration with industrials from Rhône-Alpes French region.

Acknowledgement The authors thank P. Beaune, D. Le-Berre, F. Oquendo and L. Vercouter for their contribution and useful insights.

REFERENCES

Alloui, I., S. Beydeda, S. Cimpam, V. Gruhn, F. Oquendo, and C. Schneider (2000). ”Advanced Services for Process Evolution: Monitoring and Decision Support”. In *Proc. of EWSPT*, Austria.

Alloui, I. and F. Oquendo (2001). ”Software Process Architecture = Process Components + Intention-based Interactions”. In *Proc. of PDPTA*, Las Vegas, Nevada (US).

Andreoli, J.-M., D. Arregui, F. Pacull, M. Riviere, J.-Y. Vion-Dury, and J. Willamowski (1999). ”CLF/Mekano: a Framework for Building Virtual-Enterprise Applications”. In *Proc. of EDOC*, Manheim, Germany.

Andreoli, J.-M. and S. Castellani (2001). ”Towards a Flexible Middleware Negotiation Facility for Distributed Components”. In *Proc. of DEXA ”E-Negotiations” workshop*, Munich, Germany.

Andreoli, J.-M., S. Castellani, and M. Munier (2000). ”AllianceNet: Information Sharing, Negotiation and Decision-Making for Distributed Organizations”. In *Proc. of EcWeb*, Greenwich, U.K.

Andreoli, J.-M., J.-L. Meunier, and D. Pagani (1996). ”Process Enactment and Coordination”. In *Proc. of EWSPT*, Nancy, France.

Carron, T., H. Proton, and O. Boissier (1999). ”A Temporal Agent Communication Language for Dynamic Multi-Agents Systems”. In F. Garijo and M. Boman (Eds.), *Proc. of MAAMAW*, Volume 1647 of *LNAI*, pp. 115–127. Springer Verlag.

Castellani, S., J.-M. Andreoli, and N. Glance (1998). ”Zippin: towards a graphical decision support tool”. In *CE’98*, Tokio, Japan.

Dignum, F. and U. Cortes (Eds.) (2001). *Agent Mediated Electronic Commerce III*, Number 2003 in Lecture Notes in Artificial Intelligence. Springer Verlag.

Dignum, F. and C. Sierra (Eds.) (2001). *Agent Mediated Electronic Commerce*, Number 1991 in Lecture Notes in Artificial Intelligence. Springer Verlag.

Faratin, P. (2000). *”Automated service negotiation between autonomous computational agent”*. Ph. D. thesis, Univ. of London, Queen Mary College, Dpt. of Electronic Engineering.

Hannoun, M., O. Boissier, J. S. Sichman, and C. Sayettat (2000). ”MOISE: An Organizational Model for Multi-Agent Systems”. In M. Monard and J. Sichman (Eds.), *Advances in Artificial Intelligence*, Volume 1952 of *LNAI*, pp. 156–165. Springer.

Hurwitz, S. M. (1998). W.P.: “Interoperable Infrastructures for Distributed Electronic Commerce”. http://www.atp.nist.gov/atp/98wp-ecc.htm.

Sichman, J., R. Conte, Y. Demazeau, and C. Castelfranchi (1994). ”A social reasoning mechanism based on dependences networks”. In T. Cohen (Ed.), *Proc. of ECAI*, Amsterdam, The Netherlands, pp. 182–192.

Vaughan-Nichols, S. (2002). Web services: Beyond the hype. *IEEE Computer Magazine 35*(2), 18–21.

Vercouter, L. (2000). ”A distributed approach to design open multi-agent systems”. In *2nd Int. Workshop Engineering Societies in the Agents’ World (ESAW)*.