Controlling dynamic eCommunities: Developing federated interoperability infrastructure

Lea Kutvonen

Department of Computer Science, University of Helsinki, Finland
Lea.Kutvonen@cs.Helsinki.FI

1 Introduction

Types of electronic communities vary from supply chains and CSCW environments to matrix and process organizations. Essentially eCommunities are dynamically established communities of services collaborating for reaching some joint goal. The eCommunity becomes defined by the roles of participants, their responsibilities, and expected interaction patterns among the roles for reaching the goal. Dynamicity aspects of the eCommunities go beyond the dynamic establishment phase and partner discovery during it. The community structure and the interaction patterns may be dynamic as well; a community may reflect the situations rising in its operational environment.

Regardless of the architectural style of the community, the expectation for future computing platforms is to provide a generic network infrastructure that supports these organizations.

The challenges for the networking environment can be categorized in two groups, breeding environment and operational environment. The breeding environment provides facilities for discovering potential partners and ensuring the interoperability between partners. Furthermore, the breeding environment should support agreement on the collaboration pattern to be used. In the context of virtual enterprises (or using related terms, extended enterprises, business networks) the collaboration pattern is expressed by business process models.

Earlier survey for example in IDEAS project has show that further work is required for modeling collaborative business processes. For example, ontologies for business processes are lacking and thus common understanding of the nature of collaborations is difficult to reach. In addition, the repositories of business process models should support verification of models and substitutability testing facilities. An essential but still missing aspect is the management of trust related concepts. Additional requirements for business process models rise from the need of the models and parter services to evolve.

The operational environment provides facilities for executing the collaboration, monitoring its progress, detecting breaches of collaboration agreements and running recovery or sanctioning processes due the breaches.

The alternative approaches for the operational environments cover the following; all of which are under active development.

- The operational system is built according to the business process model.
- The operational environment executes the abstract business process model.
- The operational environment allows component services to execute and monitor their conformance to the business process model requirements.

The operational environment can be constructed using either an integrated, a unified or a federated approach. Integrated approaches trust on same technical, semantical and pragmatical solutions in all enterprises. Unified solutions trust on shared metamodels supported over a heterogeneous environment. Finally, in federated solutions at leas part of the collaboration features lack a shared operational platform or metamodel.

The web-Pilarcos project follows a federated approach and provides facilities for agreeing on business process model to govern collaboration, but only as a conformity requirement. The operational environment is instrumented with sensors detecting anomalies in the system behaviour and reporting them to metalevel monitors for decision making. The corrective decisions can vary from the eCommunity dissolving or a partner leaving the community to the startup of a sanction process in addition of the normal community collaboration. The mechanisms provided are at middleware level and use reflective mechanisms. The overall system requirements are propagated to business process models as well, because some aspects of the operational environment are specific for the eCommunity itself.

The web-Pilarcos project is currently in progress of implementing some prototype facilities for B2B middleware. Thus, experiment reports are not yet topical.

In the following, the web-Pilarcos approach is described briefly. In context of Interop NoE activities on state of the art surveys, this paper shows one type of eCommunity control mechanism and business process management approach that can be used as part of the roadmapping analysis.

2 Federation management approach in web-Pilarcos

Federated communities are networks of enterprises in collaboration, using interoperable services. Our essential goal is to support dynamic collaboration between service components, across autonomic enterprises. Each collaboration is modeled by a federation contract, that explicitly but in platform-independent terms declares the functional cooperation between members, identifies the members of the federation, and captures the non-functional features related to the collaboration. This contract is used during the collaboration life-time to control the interactions within the federation, and to facilitate changes to the federation contract itself.

The collaboration pattern involves five elements. The first element captures the the service components that provide computation for reaching the goal of the collaboration. These components form a partner grid with various addressee-counterparty patterns. The second element represents an authority that supervises the interactions between the partners of the grid. This authority has a "trusted third party" role – although the responsibilities may be distributed in practice. The third element captures the language for contract specifications so that the supervising authority is able to interpret contracts. The fourth element involves the process of creating contracts using a family of contract templates; the contract template structure stems from the basic structure of the collaboration. The fifth element captures execution of the contracted computation and communication under the supervision of the controlling authority. The model is captured in Figure 1.

The management facilities of the collaboration pattern involves two levels, a real system level and a metalevel, as shown in Figure 2.

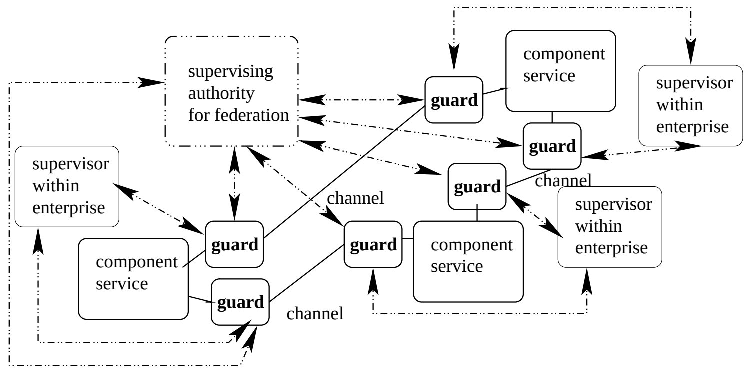

Fig. 1. The web-Pilarcos variation of supervised interaction systems.

The collaborative behaviour within the federation is monitored against the business network model that indicates the architecture of the federation. The partner roles and interactions between roles capture the relevant elements for (virtual) enterprise modelers. However, for the monitoring facilities, the structure has been supplemented with monitor elements - basically giving rules for generic monitor elements.

These monitor elements act as sensors in the real system of component services interacting with each other. Reports of occurred events are gathered to higher level monitors that are able to make deductions about the real system state and decide whether reorganizing processes should be started. The reorganizing processes use breeding environment facilities for selecting new members etc.

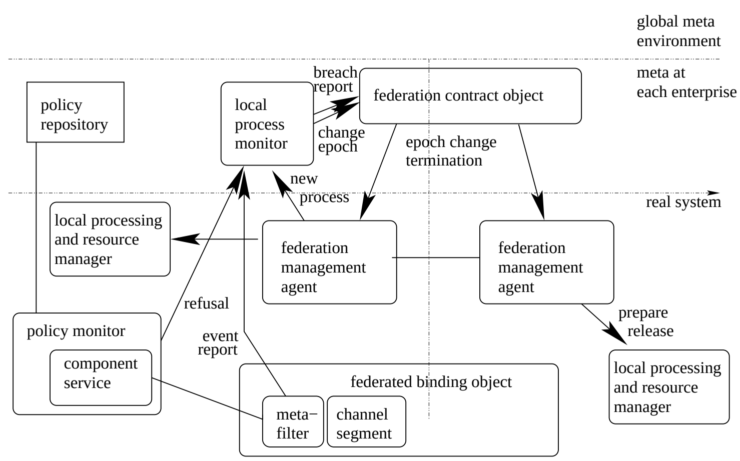

Fig. 2. Epoch changes and breach detection.

At the operational time in the federation, three flows of information are active. Fundamentally the system is about the base-level system information flows that are driven by the component services. By intention, these information flows conform to the business network model information flows, but as the components have been implemented separately, their behaviour is only believed (trusted) to be correct. Thus, a passive enforcement mechanism is needed.

The secondary flows move around metalevel information. Policy monitors and metafilters act as sensors and create meta-notifications of interesting events. These meta-notifications are delivered to local process monitor that is able to make deductions about the system state and its conformance to the expected state. In cases where corrective actions are needed the local process monitor sends requests to more powerful agents.

Triggers for this sensory flow stem from the invocation events passing either of the guards illustrated in Figure 1 or by local policy monitor. In addition, timed obligations for interaction that has not been detected by the guards, triggers a meta-notification to be delivered.

In addition to the sensory flow, a rule flow goes to the opposite direction and delivers rules for deciding what are the interesting events. Triggers for the rule flows stem from the middleware API operations or from the reflective decisions within management agents or federation contract object.

In addition to the rule flow towards already existing elements, the metalevel agents may do decisions concerning the system restructuring. For example, metalevel decisions may lead to automatic actions on for example membership changes or other operations available on the middleware API. The required changes are put in effect by predefined scripts that send messages to enterprise agents. These messages are at each enterprise separately mapped onto local service management operations, i.e., onto platform-dependent forms.

3 Federation management services

The federation life cycle consists of establishment, operation, and termination phases.

The federation establishment phase involves discovery of potential participants and ensuring of interoperability with them before committing to collaboration. The collaborating service components are aware of the business process model used between them, but need to ensure that the models referred to match to each other according to an ontology. The federation contract can also refer to non-functional commitments of component services, such as trust, security, QoS, and information representation formats. The selection of participants is made complex by two aspects: the requirement of leaving enterprises autonomic, and the requirement of ensuring interoperability in a federated environment.

Autonomy of enterprises spans a) selection of computing platform, and schedule of technical changes in it, b) selection of service components put externally available, c) evolution life-cycle of each offered enterprise application, including withdrawal of services already part of some federations, d) decisions on the kind of collaborations that are entered, and e) decisions on the kind of partners are accepted.

In the breeding environment, these aspects become visible as different metainformation elements. Those services an enterprise wishes to put available by other enterprises are exported to service repositories (such as traders or UDDI). As the selection of services changes, the set of offers is changed, thus announcing the steps of evolution within the enterprise to the potential users of the services; the service repository is consulted when new federations are formed. In operational federations, the termination of the support for a component service will cause breaches, and the breach recovery processes in turn will consult the service repository for replacement services.

The final steps of federation establishment process requests commitment to the suggested federation from all selected partners. During this step, an enterprise may by private decision refuse from participating the federation. In that case, another suggestion with a replacing service is picked. These decisions may be based on the type of collaboration suggested, or on the distrust on suggested partners.

Service offers also describe the technical processing and communication requirements in the extent that is relevant for interoperability between component services. The static analysis of interoperability between two components is based on comparison tests on the interface descriptions (including protocol descriptions) and ensuring that name given for expected technical and semantical communication constructs match.

The pragmatic interoperability cannot be statically analyzed. Instead, operational environment facilities are required. In each enterprise we expect to have a private policy repository that captures rules for accessing and distributing service processing and documents. The policy repository includes access and processing rules for the local resources (service components, documents) and potential interactions. These resource guards can be implemented in a similar style as has been presented in other policy-based management work taking deontic policies into account. Each resource (processing unit, document) is governed by a monitor that consults the local policy repository for permission to proceed with a requested interaction.

The local policies may change during the operational time of a federation. Even if changes are not propagated to existing federations, contradictions can arise during operation.

The operational phase management involves two major elements: monitoring and changes of the federation state and structure.

During the operational phase, the federation structure may change, because the members want to do changes or because of technical failures or evolution. It is possible that a) a member decides to leave a federation, b) a member is forced to leave a federation because of breaches, c) a member changes its mechanisms for providing the services it is responsible of, and d) the collaboration process reaches a new epoch where the structure of the federation is different.

The operational phase again addresses the autonomy requirements of enterprises. The enterprises may at will decide to join or leave a federation, assuming that the federation rules and other enterprises accept. Within each collaboration situations may rise where the operational goals of the collaboration and an enterprise contradict. In contradictory situations, enterprises should be autonomic in deciding whether they act according to their internal interests (and expect the sanctions of contract breaches) or comply to the federation rules. Participation in one federation does not exclude participation in another, unless federation structuring rules explicitly state so. The concurrent federation contracts can be different, even contradictory to each other from the joint member's point of view. This is rather a benefit, as the potential for establishing for example electronic commerce networks rises as the enterprises are not unduly limited by their partners computing requirements and operational policies in other collaborations.

The operational phase involves another form of dynamism as well. The business network rules may include epochs. An epoch is a block of collaboration where the set of roles and services is stable. An epoch change captures a major reorganization of the collaboration structure, membership etc.

For the management of whole federations during the operational phase the platform provides operations to a) terminate federation, b) notify of entering compensation process, c) notify on detected federation contract breach, d) query federation contract metainformation and federation status in terms of progress in the business process, membership, and breach management process definitions, and e) repopulate and negotiate an existing federation. Finally, there are operations for members to join/leave a federation role during the operational phase of the federation.

These operations are made available at each platform, regardless whether the service is actually realized locally or remotely supported. Some of the services required are actually global in nature.

4 Conclusion

Besides breeding environments for eCommunities, we also find it necessary to develop control environments for monitoring and reflectively restructuring the operational eCommunities. With these facilities, eCommunities gain in operational time dynamicity, independence of technical computing platforms, and independence of internal processing detail of collaborating partners.

This work requires further development of business process modeling techniques. First of all, collaboration of business processes or workflows should be modelled without unnecessary reveling of local processing steps. Instead, only the collaborative part – external view – should be agreed on and monitored. One line of progression here is the component-driven approach on splitting workflows into webservices or other similar entities.

The structural needs of business process models are also widened by the requirements of incorporating reusable sanctioning, recovery, and compensation processes into federation contracts.

Furthermore, shared ontologies and repositories for business process models should be made available. Such facilities would improve the potential for reaching interoperability in an environment where service components are truly developed independently from each other. More fundamentally, ontologies and repositories would create a facility for checking semantical similarity of business process model as part of the interoperability tests during eCommunity establishment.

References

1. ISO/IEC JTC1: Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. Reference Model of Open Distributed Processing. Part 3: Architecture. (1996) IS10746-3.
2. ISO/IEC JTC1: Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. ODP Enterprise Language. (2003) IS13235.
3. Kutvonen, L.: Automated management of interorganisational applciations. In: EDOC2002. (2002)