11

Cross-organizational Workflow Management

General approaches and their suitability for engineering processes

Ottokar Kulendik[1], Kurt Rothermel[1], Reiner Siebert[2]
[1] University of Stuttgart, Institute of Parallel and Distributed High-Performance Systems, Stuttgart, Germany
2 DaimlerChrysler Research & Technology, Lab. IT für Engineering, Ulm, Germany

Abstract: Cross-organizational workflow management deals with the need for transparent and controlled process automation across organizational boundaries. Cooperation between manufacturers and suppliers in the field of engineering requires coupling of parallel workflows of autonomous organization units. So far, several approaches with different architectures exist but they address this field of application differently. This paper presents some requirements of cross-organizational workflow management in the engineering domain by using an application scenario. Existing research approaches are classified distinguishing used specification schemata and expressiveness concerning cross-workflow dependencies. We describe a new approach in a class that matches the presented requirements.

1. INTRODUCTION

The situation on today's markets forces companies to reposition themselves, to rethink their core competencies, and to cooperate with each other. Increasing availability of e-business in the economic landscape is basis for automation of business processes. For established cooperation relations, permanent cost pressure increases the need for a stronger integration and even standardization of business processes, e.g. between manufacturers and suppliers. Common base for the cooperation support can be the coupling of business processes realized as workflows. This problem arises specifically where manufacturers and suppliers cooperatively develop vehicles. To achieve control over and changeability of cross-organizational development processes, they can be supported using workflow management. Important goals for the support of these engineering processes are the following ones.

Support for activity-specific participation and divided development responsibility. On the one hand, suppliers in the engineering domain may participate in specific activities of manufacturers' processes. On the other hand, as we will show below with the help of a scenario, in general suppliers may have own workflows that run parallel to manufacturers' ones. There may be multiple dependencies in between such workflows. In such cases, suppliers, like manufacturers, participate in an enormous part of a development process and perform iterations of constructive change and digital and physical validation of CAD models. They coordinate the development and are accountable for developed parts and modules in that they fulfil the requirements. Of course, due to complex dependencies between construction and validation departments in today's simultaneous engineering processes, this case is also relevant for in-house processes in large companies where individual development departments have their own processes and may even be geographically distributed. Cross-organizational workflow management should support divided development responsibility.

Integration of existing workflows and workflow management systems. Manufacturers and suppliers have their own information systems and business processes. So, instead of centralized processing and prescription of systems and processes, their integration should be achieved to maintain the autonomy of different organizations concerning information and business process management. Because of this, existing workflows and heterogeneous workflow management systems have to be integrated into cross-organizational workflows. It cannot be assumed that one workflow system prevails against others due to the increasing diversity of these systems (cf.[1]). Workflow systems in operation are more and more integrated into special purpose and application systems like e.g. enterprise resource planning, engineering data management, or enterprise application integration. A definition of new workflows should not be enforced where possible to reduce set-up effort.

Support for engineering processes. While parts of development processes are well-structured and can be planned and defined a priori, as we will see in the scenario below, process support for collaborative engineering processes has to support unstructured parts of work by allowing dynamic decisions concerning activity sequence and refinement, long-running activities and unstructured process parts as well as a coarse-grained specification[2]. These requirements are reflected differently by specific systems developed in research[2,3] or systems in operation like PDM systems with limited workflow support.

How should an approach for cross-organizational workflow management in this environment look like? To date, several research approaches for cross-organizational workflow management have been developed. Using a scenario from the engineering domain, we explore some requirements for an approach in this field of application. We describe a reference model and use it to classify existing approaches along the identified requirements. We then describe our own approach chosen in a class which matches the requirements best. After discussing related work, we conclude and give an outlook on future work.

2. REQUIREMENTS

The following scenario is a simplified illustration of a part of a development process for a vehicle's body. Background is a cooperation project with an engineering supplier that bears responsibility for a module like the side door. Figure 1 shows an early phase of the entire development process where only construction and simulation departments are involved on both sides. Depicted is a high-level view on activities visible to both manufacturer and supplier that are realized by detailed internal workflows.

2.1 Scenario

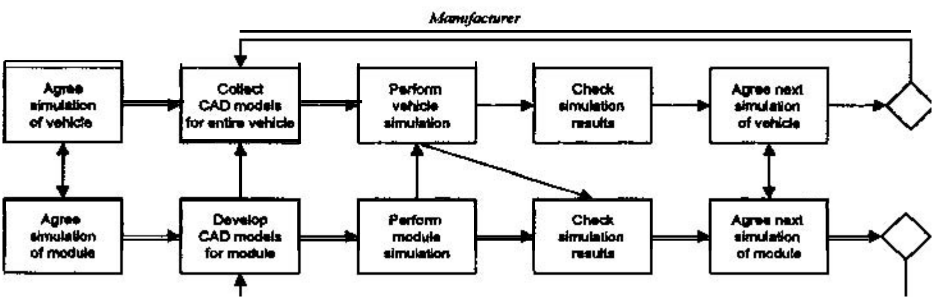

Figure 1. Scenario

The workflow is triggered at the beginning of a new development phase where some changes resulting from feedback of preceding phases have to be elaborated and the digital vehicle should fulfil some safety and quality requirement specifications that are digitally verified. At first, project leaders of development of both manufacturer and supplier come to an agreement concerning deadlines for the minor development phases up to a milestone at the end of the major development phase. In the scenario shown, these are deadlines for preparing a new version of CAD models of the module and for getting back simulation results. The supplier develops the CAD models agreed upon and transfers them to the manufacturer. Then, the supplier performs a module simulation, e.g. a dent simulation. The manufacturer receives the results and starts a simulation of the entire vehicle using the same CAD models, e.g. for a crash simulation. After simulation, results are transferred to the supplier. Meanwhile, the project leaders of development of both supplier and manufacturer evaluate results of the simulation. If a problem in one or both simulation occurs, optimisations and another iteration before the end of this development phase are necessary and are again agreed by both project leaders of development.

2.2 Requirements

Bearing this scenario in mind, we explore some requirements for cross-organizational workflow management.

Coupling of parallel workflows with multiple cross-organizational workflow dependencies. As shown in the scenario, the supplier has an own workflow that runs parallel to the manufacturer's one. In contrast to a call of a remote subworkflow with dependencies only during calling and getting back results, there are multiple control and data flow dependencies in between the workflows. This situation is typical for scenarios with divided responsibility for development because of the need for coordination and tight cooperation along aggregation or neighbourhood relationships of parts, modules and entire body of the vehicle.

Common view on cross-organizational workflows for participating organizations. Especially when responsibility for product development is divided, manufacturers need a view on suppliers' planned or running workflows, organizational structure, and data; and so do suppliers. Specifically, concerning aspects of function, information, and behaviour, specification information with respect to planned course of activities and data is important; as well as monitoring information like state of the process, etc. Regarding organizational aspects, contact persons or performers of planned, running, or past activities are important, for instance to cope with information demand concerning simulation results. This process transparency is needed due to a high degree of dependency between involved processes. It increases level of knowledge of managing as well as of operative process participants. It also achieves a better identification with the entire process and thereby finally enhances process quality. To cope with the obvious need of privacy and abstraction when integrating internal workflows, a specific view onto internal workflows is desirable to make them visible externally for specifying how workflows should be coupled and for monitoring by other organizations.

Adaptations of cross-organizational and internal workflows. After the configuration of cross-organizational workflows those workflows as well as workflows of manufacturers or suppliers have to be adapted to new project scopes and responsibilities, or to actual demands concerning safety, quality or innovation. This can result in the need for additional activities, e.g. added checks or quality assurance examinations. To keep change effort low, changes of cross-organizational workflows should not result in changes of internal workflows of organizations, and vice versa.

Complex cross-organizational control and data flow dependencies. As with intra-organizational workflows, control and data flow dependencies should be able to be restricted to certain conditions. For example, it is aspired that several control flows can be synchronized, e.g. using an AND-join or an OR-join[4], if the manufacturer has to wait for the completion of another module's CAD models by another supplier. To support overlapping activities like e.g. the agreement shown in the scenario, more complex inter-activity dependencies than just successor-relationships are desired. Furthermore, in the engineering domain, specification of dependencies between workflows with unstructured parts should be supported. For instance, if two developers of neighboured parts should level out their models, a sequence of their activities may not be defined a priori, but the existence of predefined intermediate states of workflow data.

3. GENERAL APPROACHES

In order to be able to differentiate between approaches we describe a reference model in the following that helps to derive a classification system for approaches. The model which we derive reflects the need for providing an external representation of internal workflows which was mentioned in the second requirement above. We then use the model to distinguish two criteria of existing approaches: in what kind of schema dependencies between the workflows are specified (direct/indirect coupling), and the expressiveness of the model for specifying dependencies between workflows (subworkflow/multiple dependencies).

3.1 Reference architecture

To describe and to compare different approaches for cross-organizational workflow management, we look at how these workflows are specified. Following the terminology of [5], in the following workflow schema model denotes an abstract idea of a workflow. An approach for cross-organizational workflow management may choose to specify a workflow schema model using a worflow schema, that means a workflow of its own. A schema is a specification in a formal textual or graphical language. It may be structured into different sub-schemata regarding the different workflow aspects of function (activities performed), information (use and flow of workflow data), behaviour (sequence of activities), organization (roles that perform activities), and operation (tools for performing activities). The reference architecture consists of the three entities: high-level workflow schema model, external workflow schema model, and internal workflow schema model (cf. figure 2). After describing these entities, we can distinguish between approach classes regarding how the high-level workflow schema model is specified, and how dependencies between workflows are expressed. We compare the resulting classes of approaches concerning the requirements.

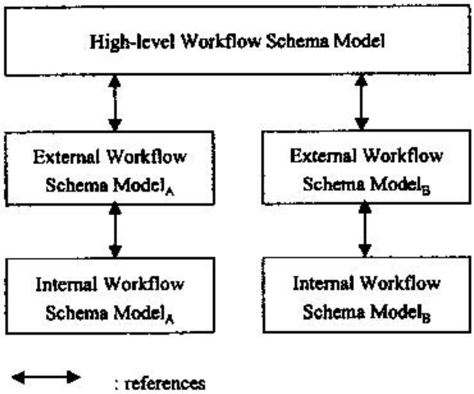

Figure 2. Reference model

The high-level workflow schema model comprises those parts of a cross-organizational workflow in which several organizations take part. Specifically, it describes which organizations take part in which workflows. Therefore, it references corresponding external workflow schema models of each organization.

The external workflow schema model describes a workflow part of a high-level workflow which belongs to an organization. It contains externally visible aspects of a workflow and so it supports encapsulation for privacy and abstraction purpose. This schema model is independent of any concrete participation in a cross-organizational workflow.

The internal workflow schema model describes an internal workflow of an organization. It consists of the full workflow specification needed for execution, e.g. aspects of function, information, behaviour, organization, and operation. Because it contains confidential details of the organization, e.g. the workflow structure and data, and unnecessary details, the internal model is just visible to the own organization. The internal workflow schema model is specified by an internal workflow schema that is managed by an internal workflow management system. The internal workflow management systems of different organizations may be heterogeneous.

In contrast to the internal workflow schema model, existing approaches do not always specify the other both schema models by workflow schemata. However, an external workflow schema model is at least conceptually separated from an internal schema model. This schema can be used to mask heterogeneous workflows and workflow systems to the high-level workflow, therefore some approaches specify further schemata in order to map between external and internal schemata or to integrate external into high-level schemata. A detailed discussion of these schemata and distributed implementation of logical components lies beyond the scope of this paper.

In the following, we first use the specification of the high-level workflow schema model as characteristic criterion for an approach's architecture. We mainly distinguish between two general approach classes, direct and indirect coupling, and some specific variants. Independent of this difference, we examine as a further criterion how dependencies between workflows of different organizations are specified.

3.2 Direct coupling vs. indirect coupling

With respect to how a high-level workflow schema model is specified externally or inside internal workflow schemata, we can distinguish between direct and indirect coupling.

Direct coupling integrates existing workflow management systems without using an additional schema for the high-level schema model (cf. figure 3). Using this approach, an organization extends its internal worflow schema by importing an external wokflow schema of another organization. So, dependencies between workflow parts of different organizations are specified inside internal workflow schemata, and a high-level workflow schema model is specified by these internal and external workflow schemata. During execution time, internal workflow management systems instantiate internal workflow schemata and manage the workflow instances. In case of control or data flow dependencies with another workflow they interact with other internal workflow management systems by using an integration layer. The topology of the interaction is according to the dependencies between the corresponding external and internal workflow schemata. Examples are ACEFlow[6], CrossFlow[7], Mokassin[8], and WAGS[9].

Direct coupling with high-level workflow schema is a hybrid form of direct and indirect coupling concerning specification schemata and execution architecture: during specification time, a high-level workflow schema is specified, during execution time the architecture corresponds to direct coupling. After specification, the high-level schema is divided into organization specific external and internal schemata. Each organization may implement generated external schemata respectively extend or refine generated internal schemata that are then executed by their internal workflow management systems. To allow for that, high-level workflow schemata are often structured by using special concepts that support an easy division of the high-level schema. Interworkflow[10] supports this approach. It provides additional support for top-down modelling of cross-organizational workflows. In the following, we consider direct coupling with high-level workflow schema as a special case of direct coupling because both classes of approaches have basically the same schemata during execution time.

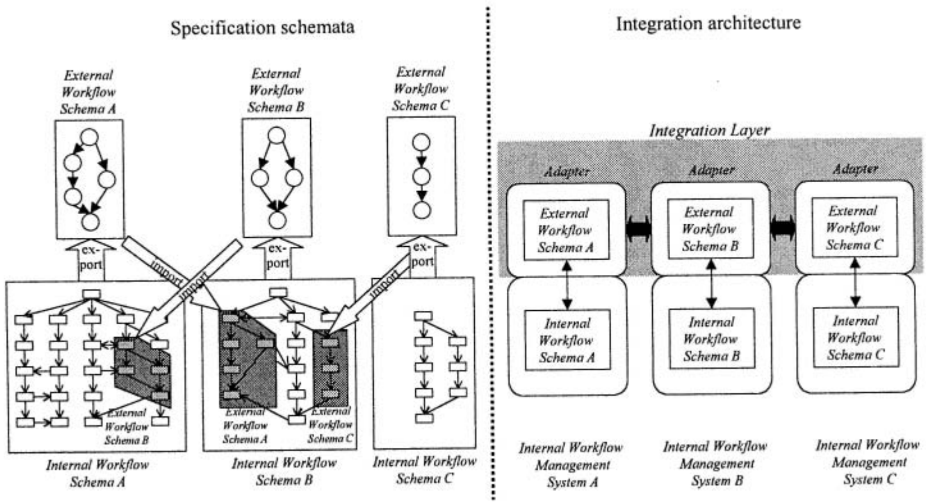

Figure 3. Direct coupling

Indirect coupling uses an additional high-level workflow schema for specifying a cross-organizational workflow. Depending on whether a high-level workflow schema is managed by a centralized or a decentralized system, we can distinguish between two variants of indirect coupling.

Centralized indirect coupling aims to support a centrally coordinated cooperation, where a high-level workflow schema model is specified and executed in an additional central workflow system. WISE[11], CMI[12], and eFlow[13] belong to this class. In this kind of approach a high-level workflow schema model is specified in a high-level workflow schema. The high-level workflow schema is managed by a high-level workflow management system, and instances of it are executed there. This system is a special workflow management system for managing cross-organizational workflows.

For specification purpose the high-level workflow management system may offer specific concepts. So, a high-level workflow schema contains references to external workflow schemata that are involved in the cross-organizational process and specifies control and data flow dependencies between them. Since it is a workflow management system on its own, it can contain own activities that are performed by members of the organization unit that operates it. The schema may further contain coupling-specific control flow constructs as multiple calls of one subworkflow, dynamic binding and some more; or it may use other shared components like an organization service covering all organizations involved.

During execution the high-level workflow management system centrally manages high-level workflow instances. It interacts with internal workflow management systems to ensure dependencies between high-level workflow parts. The interaction is hierarchical, i.e. systems of the participants just interact with the high-level system and not directly with each other. The existing internal workflow management systems are integrated with the high-level system using adapters located in the high-level system or in the internal system. The interaction is realized using one or several middleware systems.

Decentralized indirect coupling manages high-level workflow schemata by using a decentralized system that consists of high-level components which belong to each participating organization (cf. figure 4). Control and data flow dependencies are specified in a specific high-level dependency schema between external workflow schemata. So, in contrast to centralized indirect coupling, the high-level workflow schema may not contain own activities that are not defined inside one of the external workflow schemata of participating organizations. During runtime, dependencies are realized by interactions between high-level components that together form one logical high-level workflow management system, while high-level workflow state inquiries result in polling of high-level components. Adapters integrate existing internal workflow management systems with high-level components. Examples are Process Fractals[l4], referential Petri-nets[15], MariFlow[l6], and VEC[17].

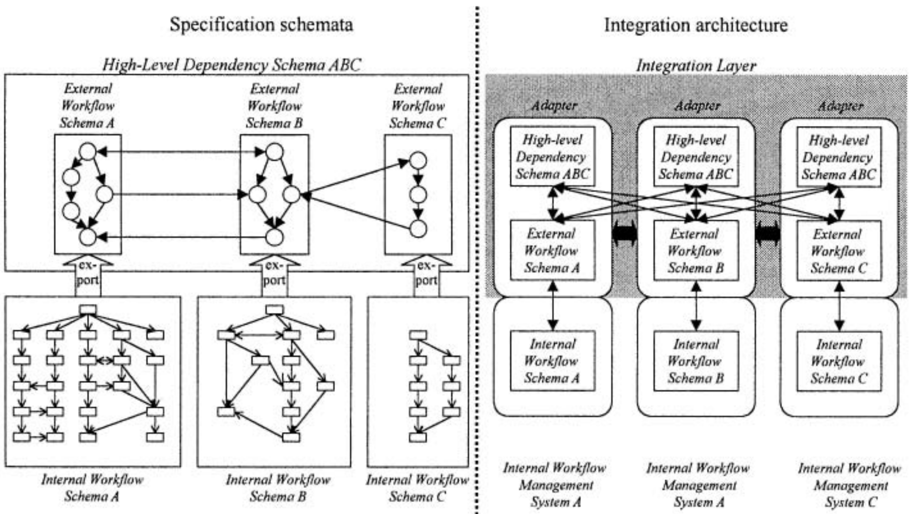

Figure 4. Decentralized indirect coupling

Comparison to relevant requirements. Indirect coupling can realize a common view using its additional high-level schemata. Direct coupling fails to support this common view well because high-level workflow schemata are specified by internal worflow schemata and are managed by internal workflow management systems that are private to organizations and should not be accessible by other organizations. In addition, a common view of a high-level workflow only using external workflow schemata as available in indirect coupling can be used simultaneously by all participating organizations because no internal workflow schema is involved.

Since indirect coupling uses an own schema to specify dependencies between external workflow schemata, it is much better in supporting adaptations of high-level workflows and internal workflows. In contrast to that, direct coupling requires a change of an internal workflow schema if a different external workflow schema has to be used, e.g. when a different organization participates.

For similar reason, indirect coupling allows specification of complex dependencies more easily than direct coupling. Because dependencies can be specified in own schemata that can be designed independent from underlying internal workflow management systems' concepts, specific dependency concepts for cross-organizational workflow management can be added more easily in these approaches as functionality can be added outside of existing systems. Though, concrete extensions may require extensions of internal systems as well and therefore their integration into specific systems have to be considered in detail. Note that despite of these issues, direct coupling has the advantage of allowing reuse of tools for monitoring of a high-level workflow easier than indirect coupling. However, this is a characteristic we do not require.

3.3 Subworkflow dependencies vs. multiple dependencies

Approaches of direct as well as of indirect coupling differ concerning how dependencies between workflows can be specified. Several approaches support *subworkflow dependencies* between two workflows [6,11,16,17]. This corresponds to outsourcing of an activity. If workflow WA has a subworkflow dependency to workflow WB, WA transfers control and data on start of WB, WB is performed, and WA gets back control and result data after completion of WB. An external workflow schema then just specifies a signature of WB for coupling. Interface 4 defined by the WfMC[18] that describes an interoperability interface for workflow management systems is usable for realizing subworkflow dependencies.

Other approaches [7-10,12-15] use more complex external workflow schemata to allow for *multiple dependencies between workflows.* These approaches describe not just input and output parameters but intermediate states and data of workflows as well. Control flow dependencies can be made conditional on these intermediate states. Input data can be communicated to a workflow which already started, output data like intermediate results can be obtained before ending. Intermediary states and their association with needed input data or output data are often specified by using events that signal state changes.

Comparison to relevant requirements. As we explained in the section on requirements, we need to support multiple dependencies between workflows of autonomous organizations. While it is in principle possible to divide workflow dependencies into several subworkflow dependencies, this would not be sufficient to cope with these requirements because it would lead to several subworkflows controlled by a calling workflow. This would not allow a called organization unit to define a workflow continuously and autonomously. Furthermore, "call-back" subworkflows and complex call sequences would be needed to return intermediate results to a calling workflow.

4. A SPECIFIC APPROACH

Our comparison between classification criteria and identified requirements suggests that an approach of indirect coupling with multiple dependencies suits our application domain best. The evaluation considered the potential of each detected class of approaches rather than characteristics of concrete instances of approaches. Especially in order to realize the requirements of providing a common view and supporting complex cross-organizational dependencies, we use expressive external workflow schemata that specify the information needed for coupling as well as for monitoring, and different types of dependency specifications.

Following the general approach of decentralized indirect coupling, we propose the following architectural model (see figure 5). From bottom to top: the internal workflow layer contains internally defined and managed workflow schemata. The external workflow layer contains external workflow schemata used for coupling and for monitoring. The layer also can specify a mapping between internal and external workflow schema, which is not further described in the scope of this paper. The linkage specification describes how workflow dependencies are bound to the external workflow schema. The workflow dependency layer specifies with a configuration schema control and data flow dependencies of the workflows that together form a high-level workflow. In addition to the mentioned schemata, the layers contain the corresponding instances as well.

The external workflow schema describes externally visible aspects of function, information, behaviour, and organization. An operational aspect is not described. The functional aspect describes activities performed by the workflow and input and output relations to data objects of the informational aspect. The informational aspect describes workflow-relevant data, i.e. control data objects and production data objects. Workflow data objects are described by name, type, and location. Analogously, the organizational aspect is realized by reference to a (potentially external) organization service. In addition to roles and persons that perform activities, contact persons for an activity are referenced. For all aspects, schema as well as instance information is available, e.g. the organizational aspects allow us to retrieve which person performed a certain activity, and so on.

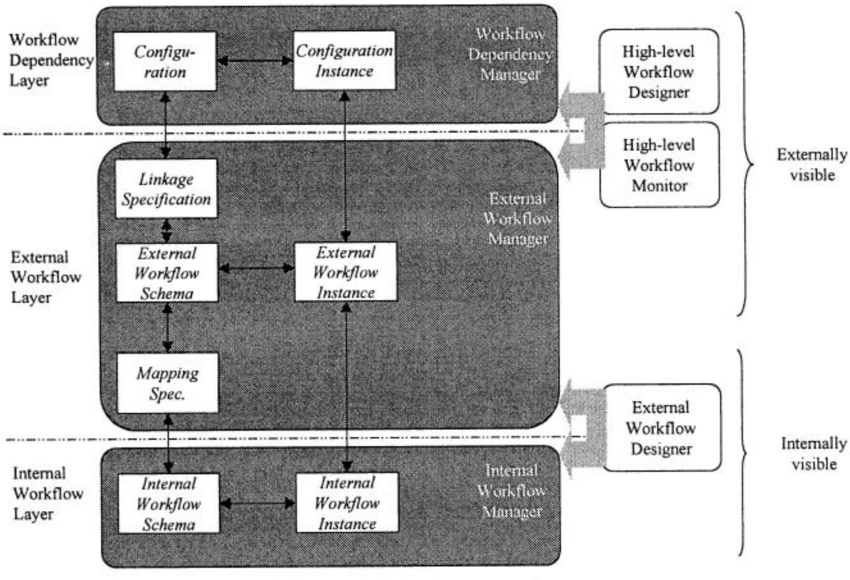

Figure 4. Proposed solution approach

The dependencies between the participating organizations are defined using a common configuration schema that specifies the dependencies, dependency endpoints that are assigned inside the external workflow schema, and the linkage specification between those schemata. In the *linkage specification,* dependencies are bound to dependency endpoints. For example, a simple data flow dependency is bound to a data flow exit of a sending organization's workflow and a data flow entry of a receiving organization's workflow. This allows for a decentralized and autonomous specification: an organization can specify on its own which activities consume the output of an agreed data flow dependency, and it can change this specification, e.g. if a newly inserted activity should get the output data instead.

Dependencies are multiple relations that can be bound to dependency endpoints, i.e. entries and exits. Control flow dependencies describe constraints concerning execution behaviour of workflows, activities, or control constructs. A control flow dependency is connected to one or more control exits and one entry, and may contain a condition. A condition specifies in terms of the exits' states in which case the entry is activated, e.g. to model a control flow join using a logical AND. Data flow dependencies model the transfer of data between different workflows. A data flow dependency connects a data exit to a data entry. It is evaluated in case of a data state change of an exit. Data flow dependencies specify one or more assignments between data content items of data exits and data content items of data entries. Dependencies may further be associated with conditions which describe in what case data is transferred.

Dependency endpoints are part of the external workflow schema. They specify endpoints of the workflow, i.e. entries and exits of control and data flow. Sources for control and data dependency points specified in the external workflow schema may be control construct execution states or workflow data states, or application-specific events. Control flow entries and exits can be assigned to specific activity and control construct execution states, e.g. started or finished, with the semantics that the activity or control construct state can be triggered respectively delayed by a control flow dependency. Data flow entries are associated with activity input data or workflow input data states, data flow exits can be associated to activity output data or states of workflow output data. If a structure of an external workflow schema is not fully specified due to an unstructured or previously unforeseen course of activities, changes of workflow data may be used as control flow exits. In addition, application-specific input and output events may be used as control or data flow entries and exits. They can be assigned to parts of an external workflow schema later. Together, this allows us to model dependencies even where a course of activities is not known during dependency modelling time, but needed data input or produced output or some intermediate events are known in advance.

5. DISCUSSION OF RELATED WORK

As described in the section on classification, several approaches focus on subworkflow dependencies between cross-organizational workflows. Some workflow approaches focus on distribution, availability, and heterogeneity of workflows. Work dealing with similar issues as this paper engages in is Process Fractals, Mokassin, CMI, and referential Petri-nets.

Process Fractals[14] follow decentralized indirect coupling. In that approach, the external workflow schema is divided into an event-based interface used for coupling and a specific representation used for monitoring. Cross-organizational control flow is modelled by using binding input events of one interface to output events of another and vice versa. Complex cross-workflow dependencies as e.g. AND-joins are not supported. Data flow is bound to these control flow transitions and is realized by moving or copying documents that have a commonly known type. The organizational aspect concentrates on large-scale aspects like dependencies between organizations; monitoring of small-scale organizational information is not aimed at.

Mokassin[8] supports direct coupling with multiple dependencies using a decentralized agent-based infrastructure. Here, specification does not focus on separating internal and external representation but on separating interface and implementation of a workflow. With this, the approach does not aim on supporting organizational information for monitoring purpose or a separated external view on an internal workflow. An interface representation may contain a state graph specifying events that an implementation generates. Complex control and data flow dependencies are supported and can be added by a modeller.

CMI[12] supports centralized indirect coupling using a high-level workflow schema in which workflows of different organizations are imported as specific activities. The external workflow schema contains a state chart where possible interactions with the service are specified by methods that can either be called on such external workflows. Specific concepts of the high-level workflow management system are e.g. activities that can perform multiple calls on external workflows.

Referential Petri-nets[15] focus on Petri-net based modelling of cross-organizational workflows. Inheritance concepts between external and internal workflow schemata are developed, to support for controlled compatible extensions.

6. SUMMARY AND OUTLOOK

Cross-organizational workflow management for cooperation in the field of engineering has to support situations characterized by activity-specific participation and divided responsibility of development, integration of existing workflows and their management systems, and unstructured process parts. As we explained by using the scenario, for the support of cross-organizational engineering process we require: 1) Coupling of parallel workflows with multiple dependencies, 2) Common view on cross-organizational workflows for participating organizations, 3) Adaptations of cross-organizational and internal workflows, and 4) Complex control and data flow dependency specifications.

To examine existing approaches, we classified them along used specification schemata, complexity of the cross-organizational dependencies, and execution architectures. Some approaches are not that suitable to support adaptations of cross-organizational and internal workflows because they require change of internal workflows when cross-organizational workflows change; or participating organizations do not have a common view on cross-organizational workflows. Other approaches have just a simple subworkflow-based dependency model that makes it difficult to couple parallel workflows.

The favoured architectural class couples existing workflow management systems directly but specifies cross-organizational workflow dependencies in a specific specification and execution layer. The approach we have proposed showed how the selected architectural class can fulfil the identified requirements by using an expressive external workflow representation. Furthermore, it supported complex control and data flow dependencies across organizations. To avoid changes of the dependency schema, these dependencies can be linked to the workflows decentralized and autonomously by each participating organization.

Right now, we are detailing the model and will refine the architectural specification. To assure integration with existing systems, we are going to examine those systems to derive at a generic workflow meta schema and architecture. This provides the base for developing a detailed integration approach. In addition, issues like the support for change of schema and compatibility, and specific security requirements will further be evaluated.

7. REFERENCES

[1] Bussler C. Enterprise-wide workflow management. IEEE Concurrency 1999; 3:32-43.
[2] Reinert J., Ritter N. Applying ECA-Rules in DB-based Design Environments. Proceedings of the GI-Fachtagung CAD'98; 1998 Mar 5 - 6; Darmstadt. Informatik Xpress 9. Bonn: Gesellschaft für Informatik [u.a.], 1998.
[3] Beuter T., Schneider P., Dadam P., The WEP Model: Adaptive Workflow-Management for Engineering Processes. Proceedings of the European Concurrent Engineering Conference (ECEC'98); Erlangen-Nürnberg, 1998 Apr 26 – 29. Society for Computer Simulation; 1998.
[4] Workflow Management Coalition. Terminology & Glossary. WFMC-TC-1011, 3.0, 1999. http://www.wfmc.org/.
[5] Jablonski S., Böhm M., Schulze W. (edts.) Workflow-Management: Entwicklung von Anwendungen und Systemen. Heidelberg: dpunkt-Verlag, 1997.
[6] Stricker C., Riboni S., Kradolfer M., Taylor J. Market-based Workflow Management for Supply Chains of Services. Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS-33); Maui, Hawaii; 2000 Jan 4 – 7. IEEE Computer Society Press, 2000.
[7] Hoffner Y., Ludwig H., Gülcü C., Grefen P. Architecture for Cross-Organisational Business Processes. Proceedings of the Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS2000); 2000 Jun 8 - 9; Milpitas. IEEE Computer Society, 2000.
[8] Gronemann B., Joeris G., Scheil S., Steinfort M., Wache H. Supporting Cross-Organizational Engineering Processes by Distributed Collaborative Workflow Management - The MOKASSIN Approach. Proceedings of the 2nd Symposium on Concurrent Multidisciplinary Engineering (CME'99) / 3rd International Conference on Global Engineering Networking (GEN'99); 1999 Sep 14 - 15; Bremen. Bremen: Hochschule Bremen, Institut of Aerospace-Technology - Reports, 1999.
[9] Riempp G., Nastansky L. From islands to flexible business process networks – enabling the interaction of distributed workflow management systems. Proceedings of the 5th European Conference on Information Systems, Vol. 1; 1997 Jun 19 - 21; University College Cork, Cork, Ireland. Cork: Cork Publ. Ltd, 1997.
[10] Hiramatsu K., Okada K.-I., Matsushita Y., Hayami H. Interworkflow System: Coordination of Each Workflow System among Multiple Organizations. Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems (CoopIS'98); 1998 August 20 - 22; New York City. IEEE Computer Society Press, 1998.
[11] Lazcano A., Alonso G., Schuldt H., Schuler C. The WISE approach to Electronic Commerce. International Journal of Computer Systems Science & Engineering, 2000; 5:343-355.
[12] Schuster H., Baker D., Cichocki A., Georgakopoulos D., Rusinkiewicz M. The Collaboration Management Infrastructure. Proceedings of the 16th International

Conference on Data Engineering (ICDE'2000); San Diego, USA, 2000 February 29 - March 3. IEEE Computer Society Press, 2000.

[13] Casati, F., Ilnicki, S., Jin, L., Shan, M.-C. An Open, Flexible, and Configurable System for E-Service Composition. Tech. report, HP Laboratories Palo Alto, HPL-2000-41, 2000.

[14] Lindert F. Modelling inter-organizational processes with process model fragments. Proceedings of the GI-Workshop Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications, Informatik'99; 1999 October 5 - 9, Paderborn. Ulm: Universität Ulm, Ulmer Informatik-Berichte, 1999.

[15] Van der Aalst W.M.P, Moldt D., Wienberg F. Enacting Interorganizational Workflows using Nets in Nets. Proceedings of the 1999 Workflow Management Conference; 1999, November, 9; Münster: University of Münster, 1999.

[16] Dogac A., Beeri C., Tumer A., Ezbiderli M., Tatbul N., Icdem C., Erus G., Cetinkaya O., Hamali N. MariFlow: A Workflow Management System for Maritime Industry. In: Guedes Soares, C., Brodda, J. (edts.): Application of Information Technologies to the Maritime Industries, Edicoes Salamandra, Lisbon, 1999.

[17] Ludwig H., Whittingham K. Virtual Enterprise Coordinator – Agreement-Driven Gateways for Cross-Organisational Workflow Management. Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration (WACC '99); 1999 February 22 - 25; San Francisco. ACM Software Eng. Notes, Volume 24, Number 2, 1999.

[18] Workflow Management Coalition. Interoperability Abstract Specification. WFMC-TC-1012, 1.0, Oct. 1996. http://www.wfmc.org/.