A Collaboration Pattern Model for Virtual Organisations

Nikos Papageorgiou[1], Yannis Verginadis[1], Dimitris Apostolou[2],
Gregoris Mentzas[1]

[1]Institute of Communications and Computer Systems,
National Technical University of Athens
9 Iroon Polytechniou Str., Athens, Greece
{npapag@mail.ntua.gr, jverg@mail.ntua.gr, gmentzas@mail.ntua.gr}
[2]Informatics Department, University of Piraeus
80 Karaoli & Dimitriou Str., Piraeus, Athens, Greece
{dapost@unipi.gr}

Abstract. Collaboration, either inter or intra-organization is a critical business function that demands skills and knowledge spanning a wide range of domains including social, business and technical domains. Collaboration patterns can be a means to capture best practices about recurring collaborative problems and solutions. Many forms of patterns have been proposed addressing specific aspects of collaboration such as business interaction design and workflow. In this paper we propose a novel collaboration pattern model that aims to constitute a framework capable to describe recurring collaborative activities taking place in the context of virtual organisations.

Keywords: collaboration model, collaboration patterns

1. Introduction

Organisations increasingly seek to establish collaborations in order to expand their activities by forming Virtual Organisations (VOs). A VO constitutes a collaborative business environment with increased needs for modelling, executing and monitoring dynamic collaborations. In an effort to address these needs, we introduce the concept of patterns in VOs as a means for capturing and re-using recurring segments of work or parts of collaboration.

The concept of patterns is inspired by the way experts tackle work on a particular problem: It is unusual to tackle it by inventing a new solution that is completely different from existing ones. Instead, they often recall a similar problem they have already solved, and reuse the essence of its solution to solve the new problem. This kind of ‘expert behaviour’ is a natural way of coping with many kinds of problems [8]. The concept of design pattern was first introduced in the field of engineering by Christopher Alexander, a professor of Architecture in University of California, Berkley. His book, where he describes a language for architectural patterns [1], is seen as the prototype for patterns in many other domains, including Software Engineering [13] and Human Computer Interaction (HCI) [6].

We focus on applying the concept of patterns in collaborative work taking place in the context of VOs. A common definition of collaboration is: people or organizations working jointly with others or together especially in an intellectual endeavour[1] that is creative in nature[2]. In the context of a VO, many challenges encountered, both in its day-to-day operations and in its strategic choices, are often addressed collaboratively by the participating partners. The overall aim of using collaboration patterns (CPats) in the VO domain is to enhance support of the networked enterprises by providing a means for capturing and reusing solutions to collaborative processes.

In this paper we set out to develop a CPat model that aims to constitute a framework capable to describe recurring collaborative activities taking place in the context of VOs. We base our work on existing research in patterns and on our analysis of requirements of specific case studies. In the next sections we present related work and we describe the proposed CPat model consisting of a pattern structure, a categorization scheme and a diagrammatic description for VO CPats.

2. Related Work

To our knowledge, the application of patterns in VO collaboration is a new research domain. In our effort to reuse related previous research we have investigated a widespread range of related fields. Starting from the field of communities of practice (CoP), we see that CPats have been proposed [9] as conceptual structures that capture socio-technical lessons learnt in optimizing collaboration processes.

In the field of workflow systems, workflow patterns can be viewed as patterns addressing the structured part of collaborative processes. A workflow is often defined as a collection of tasks, performed by systems or humans, organized to accomplish some business process [14]. Control flow [28], resource [25] and data patterns [24] are subcategories of workflow patterns. Similar research is that of business process patterns [20], [3]. Activity theory on the other hand, provides a framework that addresses the ad-hoc part of collaboration. Activity patterns have been proposed to "formalize the structure and content of an activity and the integration methods it depends on, thereby making it reusable as a template in future activities" [17]. The Unified Activity Modeling methodology [18] is based on activity theory and aims to design systems that utilize repetitive activities [15].

In the field of Collaboration Engineering (CE) which deals with the design, modelling and deployment of repeatable processes for recurring collaborative tasks [7], thinkLets aim to describe elementary, atomic group processes from a leader's point of view. CE processes consisting of thinkLets can provide ready-to-apply facilitation "recipes" that capture the best practices of highly experienced facilitators. CE researchers have codified a large number of patterns [7], [10], [19].

[1] Collaborate, Merriam-Webster's Online Dictionary, 2007

[2] Collaboration, Oxford English Dictionary, Second Edition, (1989). (Eds.) J. A. Simpson & E. S. C. Weiner. Oxford: Oxford University Press.

Patterns have also been used to support distributed software engineering processes [22]. Similarly, service interaction patterns [2] are applied to the service composition or lower layers (message handling and protocols) in service-based business process interconnections.

A significant aspect of CPats is that of capturing knowledge exchanges taking place during collaborations. Pattern-based knowledge workflows can enable the automation of knowledge flows across an organization [26]. Patterns for knowledge management processes have also been suggested [23].

All previous mentioned research streams propose prescription-oriented patterns. Techniques stemming from Social Network Analysis have been proposed to identify and mine patterns of human interaction captured during the usage of collaborative systems. They aim to improve business processes by mining the deviations between a given model and its real world execution [12] or enable the automatic generation of abstract form of CPAts [4].

Alexander [1] suggests that, if a pattern author is not able to draw a picture, then the solution described is not a pattern. Although Alexander was referring to architectural design patterns, the necessity for a schematic representation of patterns exists also for all types of CPats. For instance, in the field of action patterns, EMOO diagrams have been proposed [5], a notation that is based on the MOO notation [16]; in workflow patterns, YAWL, a Petri-Net based graphical modelling language has been proposed [28]. Similarly, the BEMN notation has been proposed in the field of business events and event patterns [11] and the CIAN notation has been proposed in the area of interaction patterns for CSCW systems [21]. In all cases, the CPat diagram provides a schematic overview of the solution the CPat prescribes in order to facilitate the design process and reduce the likelihood of errors.

3. Collaboration Pattern Model

We define CPats as follows [29]: "*A collaboration pattern is a prescription which addresses a collaborative problem that may occur repeatedly in the environment. It describes the forms of collaboration and the proven solutions to a collaboration problem and appears as a recurring group of actions that enable efficiency in both the communication and the implementation of a successful solution. The collaboration pattern can be used as is in the same application domain or it can be abstracted and used as a primitive building block beyond its original domain.*"

From the CPat definition we deduce that CPats should prescribe a collaboration situation or problem as well as proven solutions to the problem. In addition to the concepts deduced from the definition, a CPat model should encapsulate some of the key findings of the related research in patterns, such as: (a) CPats can serve different objectives or functions and can be of different levels of abstraction; (b) CPats should trigger human and machine processes when certain events occur and condition hold [9]; (c) patterns may include a structure (in the form of a series of steps and the applicable user roles), content (describing the activities to be done), and methods for accessing IT resources to get things done [15]; (d) CPats should contain a diagrammatic description. In the remaining of this section we elaborate how these issues are represented explicitly in the proposed CPat model.

3.1 Collaboration Patterns Categorization

Related literature distinguishes two categorization approaches for patterns: according to their objective/function and according to their granularity. Virtual community patterns [9], CE patterns [7], workflow patterns [27], resource patterns [24], data patterns [24] and subcategories of knowledge-flow elementary patterns or knowledge-flow advanced patterns [26] are examples of pattern modelling approaches that use pattern categorization based on objective or function. Such a categorization is relevant for CPats because it allows e.g., identifying that two or more CPats present the same functional behaviour and hence both of them could be used in a specific situation.

In our CPat model, we propose a three-level categorization of CPats based on the VO objective or function they support. Specifically: (i) the *Strategic CPat* category contains CPats that facilitate strategic objectives of a VO which may refer e.g., to the actual VO formation, to the addition or removal of a new partner in the VO, to the structure of the VO, to actions that determine the VO goals, etc. (ii) the *Business CPat* category contains CPats that serve operational functions of the VOs (e.g. proposal preparation, conflict resolution, scheduling a meeting etc.), and (iii) the *Simple CPat* category which includes patterns that focus on elementary collaborative activities (e.g. completing a task) and are expected to propose a solution that involves a small number of simple activities.

3.2 Collaboration Patterns Structure

The CPat structure comprises all attributes needed for specifying what a specific CPat does, where it is applicable and under which circumstances it may be initiated. The CPat example provided in Table 1 model the structure of “Schedule a meeting”, a recurring collaboration activity, which takes place in the context of a VO that consists of a number of pharmaceutical companies intending to develop a drug according to a new set of drug administration regulations. According to this, a meeting needs to be scheduled either periodically (every 3 months) or upon the delay of a specific deliverable (triggers). At least 3 collaborators should be available along with the coordinator and the necessary budget (pre-conditions). When the aforementioned pre-conditions and triggers are satisfied, the specific CPat is applied and the solution it prescribes, in terms of action to be taken and tools to be used, can be followed. Specifically, the work package leader should be notified in order to send an e-mail to the VO coordinator asking him or her to notify the VO members about the need of organizing a meeting. Further, a web tool for agreeing on the meeting date is identified and possibly automatically configured (e.g., with partner names and possible meeting dates) and invoked. When the meeting takes place, the meeting minutes are stored. In case the delayed deliverable that triggered this meeting is submitted by the responsible partner before the actual meeting takes place, an exception event is triggered and the CPat is stopped; moreover, a CPat for postponing the meeting is invoked.

Table 1. VO Collaboration pattern model structure

Field	Description	*Example*
Name & No:	A name and a number for quick referencing.	*<Schedule a Meeting> - CPat 1*
Category:	Positioning in one of the proposed categories (Strategic/Business/Simple)	*Business Pattern*
Problem:	A description of the problem(s) the CPat has addressed before or it is expected to address in the future.	*Check the work progress for a specific deliverable*
VO lifecycle phase:	One or more VO lifecycle phases where it can be applicable (pre-creation, creation, operation or termination).	*VO Operation*
Application Area:	Declares the sector (e.g. Manufacturing) where it is applicable.	*Pharmaceutical Industry*
Pre-Conditions:	The list of the states and conditions that must be satisfied before the specific CPat can be considered applicable.	*(No. of collaborators > 3) AND (VO coordinator available) AND (Budget available)*
Triggers:	Events and event patterns that can trigger the execution of the CPat.	*(The last progress meeting was held 3 months ago) OR (Deliverable Derma1 delayed)*
Triggers of Exceptions:	Events that can raise an exception during the implementation of CPat.	*Deliverable Derma1 has just been sent.*
Roles:	Includes the collaboration roles that are to be involved in the CPat.	*VO Coordinator, WP leader, VO member*
Input Information:	Documents or data that will be used in terms of this CPat.	*Project's DoW, VO members' contact details*
Output Information:	Documents or data that will be produced in terms of this CPat.	*Meeting minutes document*
Duration:	The acceptable time frame in which the proposed by the CPat solution can be successfully implemented.	*2 Weeks (this can be a variable)*
Exception:	A description of an exception to the pattern (e.g. termination of the specific CPat and execution of another one).	*<Postpone Scheduled Meeting CPat>*
Post-Conditions:	Conditions and states that hold after the successful termination of the CPat.	*Meeting took place AND Agreed minutes stored in system.*
Related CPats:	- CPats that optionally can be executed in parallel or after its termination. - Alternative CPats that can be used instead of the described one. - Conflicting CPats that cannot be executed concurrently with the described one.	*-CPat that optionally: <Schedule conference call CPat>* *-CPat that can be executed subsequently: <Postpone Scheduled Meeting>* *-Alternative CPat : <Schedule an Online Meeting CPat>*

Field	Description	*Example*
Solution:	Comprises prescriptions of solutions to the designated problem in the form of action lists, workflows or even instructions for tool usage.	
	***Action list**:* *-WP leader sends email to the coordinator* *-Coordinator notifies collaborators about the need of organizing a meeting* *-Use tool for agreeing on the date* *-WP leader stores meeting minutes*	***Usage of tools**:* *- Engage with www.doodle.ch to find date based on collaborators' availabilities* *-Use a file server to store meeting minutes*

3.3 Collaboration Patterns Diagrammatic Description

We propose the Business Process Management Notation (BPMN) [30] as the notation for CPat diagrams. Figure 1 describes the BPMN diagrammatic description of the CPat "Schedule a Meeting" with a BPMN diagram. The requirements for the selected CPat graphical notation are imposed by the elements of the presented CPat model which, to our opinion, are adequately covered by BPMN because BPMN is typically used for business process modelling and has symbols that are able to model, beyond the collaborative process of the CPat solution, CPat triggers, pre and post-conditions, exceptions, input and output information. The capability of BPMN to model abstract processes is also very important. The abstract form of CPat diagrams can be used where the solution of a CPat should have the form of a loose action list and thus allow the CPat designer to decide about the rigidity of the solution that a specific CPat proposes.

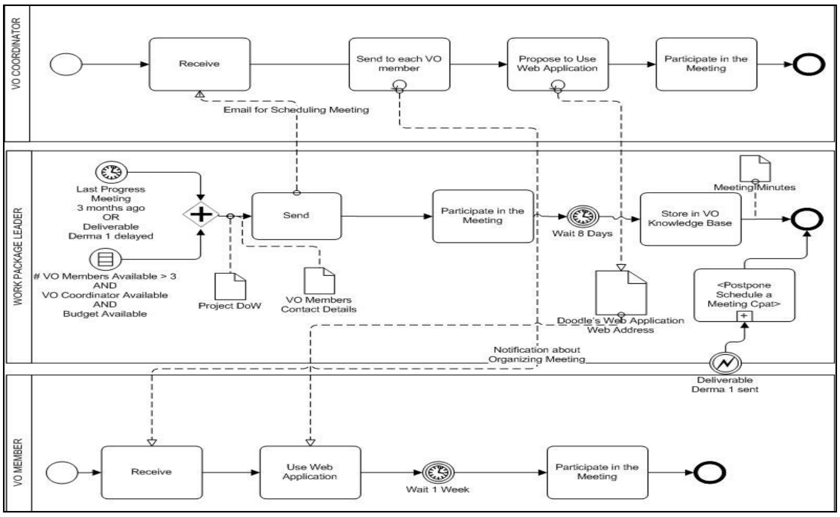
VO COORDINATOR
Receive
Send to each VO member
Propose to Use Web Application
Participate in the Meeting
Email for Scheduling Meeting
WORK PACKAGE LEADER
Last Progress Meeting 3 months ago OR Deliverable Derma 1 delayed
# VO Members Available > 3 AND VO Coordinator Available AND Budget Available
Send
Project DoW
VO Members Contact Details
Participate in the Meeting
Wait 8 Days
Store in VO Knowledge Base
Meeting Minutes
<Postpone Schedule a Meeting Cpat>
Doodle's Web Application Web Address
Notification about Organizing Meeting
Deliverable Derma 1 sent
VO MEMBER
Receive
Use Web Application
Wait 1 Week
Participate in the Meeting

Figure 1. Diagram of "Schedule a meeting" collaboration pattern

4. Conclusions and Future Work

To our knowledge, existing approaches address only isolated aspects of virtual enterprise collaboration with patterns (e.g. workflows, knowledge, ad-hoc activities). In this paper we have introduced the notion of collaboration patterns in virtual organisations. We have described a model which consists of a pattern structure and a diagrammatic description along with a complete example. This model will serve as a framework for VO CPats. With the proposed CPat model we aim to address the specific requirements of VO collaboration. We have found that the model adequately represents a number of CPats we have found in the literature and derived from case studies of the EU research project SYNERGY. The model will be used in the development of a VO collaboration platform following an event-driven architecture. The collaboration platform shall provide services for detecting and executing relevant CPats. The detection of a relevant CPat will be automatically performed upon events that match the CPat triggering conditions and provided additional conditions (problem, context, pre-conditions) hold. The execution of the CPat will be handled by appropriate collaboration and workflow services of the collaboration platform or even by humans, as prescribed in the CPat solution.

Acknowledgments. This work has been partially funded by the European Commission in project SYNERGY ICT 63631.

References

1. Alexander, C., S. Ishikawa, et al. (1977). A Pattern Language. New York, Oxford University Press.
2. Barros, A. P., M. Dumas, et al. (2005). "Service Interaction Patterns: Towards a Reference Framework for Service-based Business Process Interconnection."
3. Barros, O. H. (2004). "Business Information System Design Based on Process Pattern and Frameworks." Industrial Engineering Department, University of Chile. Available: http://www. BPtrends. com.
4. Biuk-Aghai, R., S. J. Simoff, et al. (2005). From Ad-hoc to Engineered Collaboration in Virtual Workspaces. Eleventh Americas Conference on Information Systems, Omaha, NE, USA.
5. Biuk-Aghai, R. P. (2003). An Information Model of Virtual Collaboration. Proceedings of the 2003 IEEE International Conference on Information Reuse and Integration, Las Vegas, Nevada, USA, IEEE SMC.
6. Borchers, J. O. (2001). "A pattern approach to interaction design." AI & Society 15(4): 359-376.
7. Briggs, R. O. (2003). "Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems." Journal of Management Information Systems 19(4): 31-64.
8. Buschmann, F., R. Meunier, et al. (1996). Pattern-oriented software architecture: a system of patterns, John Wiley & Sons, Inc. New York, NY, USA.
9. de Moor, A. (2006). Community Memory Activation with Collaboration Patterns. Proceedings of the 3rd International Community Informatics Conference (CIRN 2006) Prato Italy, pp. 1.

10. de Vreede, G. J. and R. O. Briggs (2005). Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks. Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)-Track.
11. Decker, G., A. Grosskopf, et al. (2007). A Graphical Notation for Modeling Complex Events in Business Processes.
12. Dustdar, S. and T. Hoffmann (2007). "Interaction pattern detection in process oriented information systems." Data & Knowledge Engineering 62(1): 138-155.
13. Gamma, E., R. Helm, et al. (1995). Design patterns: elements of reusable object-oriented software, Addison-Wesley Reading, MA.
14. Georgakopoulos, D., M. Hornick, et al. (1995). "An overview of workflow management: From process modeling to workflow automation infrastructure." Distributed and Parallel Databases 3(2): 119-153.
15. Geyer, W., M. J. Muller, et al. (2006). "ActivityExplorer: Activity-Centric Collaboration from Research to Product." IBM Systems Journal 45(4): 713-738.
16. Hawryszkiewycz, I. T. (2000). Analysis for Cooperative Business Processes. Proceedings of the Fifth Australian Workshop on Requirements Engineering.
17. Hill, C., R. Yates, et al. (2006). "Beyond predictable workflows: Enhancing productivity in artful business processes." IBM Systems Journal 45(4): 663-682.
18. IBM. (2005). "Unified Activity Management project." from http://domino.research.ibm.com/comm/research_projects.nsf/pages/uam.index.html.
19. Kolfschoten, G. L., R. O. Briggs, et al. (2006). "A conceptual foundation of the thinkLet concept for Collaboration Engineering." International Journal of Human-Computer Studies 64(7): 611-621.
20. Malone, T. W., K. Crowston, et al. (2003). Organizing Business Knowledge: The Mit Process Handbook, MIT Press.
21. Molina, A. I., M. A. Redondo, et al. (2006). "A Conceptual and Methodological Framework for Modeling Interactive Groupware Applications." LECTURE NOTES IN COMPUTER SCIENCE 4154: 413.
22. Norta, A., M. Hendrix, et al. (2006). A Pattern Repository for Establishing Inter-organizational Business Processes, Beta, Research School for Operations Management and Logistics.
23. Qureshi, S., V. Hlupic, et al. (2004). "On the Convergence of Knowledge Management and Groupware." LECTURE NOTES IN COMPUTER SCIENCE: 25-33.
24. Russell, N., A. H. M. ter Hofstede, et al. (2004). Workflow Data Patterns.
25. Russell, N., A. H. M. ter Hofstede, et al. (2005). Workflow Resource Patterns, Beta, Research School for Operations Management and Logistics.
26. Sarnikar, S. and J. L. Zhao (2007). "Pattern-based knowledge workflow automation: concepts and issues." Information Systems and E-Business Management: 1-18.
27. van der Aalst, W. M. P., A. H. M. Hofstede, et al. (2003). "Business Process Management: A Survey." LECTURE NOTES IN COMPUTER SCIENCE: 1-12.
28. van der Aalst, W. M. P. and A. H. M. ter Hofstede (2005). "YAWL: yet another workflow language." Information Systems 30(4): 245-275.
29. Verginadis, Y., D. Apostolou, et al. (2009). Collaboration Patterns in Event-Driven Environment for Virtual Organisations. Intelligent Event Processing – Association for the Advancement of Artificial Intelligence (AAAI), Spring Symposium 2009,. Stanford, USA.
30. White, S. A. (2004). "Business Process Modeling Notation (BPMN) Version 1.0." Business Process Management Initiative, BPMI. org, May 1.