

Modelling Dynamics in Collaboration: An Extension to the Collaborative Network Relationship Analysis

Heiko Duin, Jens Eschenbächer, and Klaus-Dieter Thoben

BIBA – Bremer Institut für Produktion und Logistik GmbH,
Hochschulring 20, D-28359 Bremen, Germany
{du,esc,tho}@biba.uni-bremen.de

Abstract. This is a concept paper which introduces a dynamic extension to a collaborative network relationship analysis approach. The extension is based on the MetaMatrix approach known from the Dynamic Network Analysis field. Several entity classes representing agents, tasks, resources and knowledge are introduced and possible relations between entities of the different classes are analysed. All entity classes and relation classes are attributed with time related data which allows dynamic changes the system. Finally, some illustrative examples of typical collaborative interactions are introduced and explained.

Keywords: Dynamic Collaborative Network Relationship Analysis (DCNRA), MetaMatrix, Modelling and Simulation.

1 Introduction

The Social Network Analysis (SNA) is an appropriate method to capture and analyze relations between individuals or organizations for a given point in time (Wasserman & Faust, 2007). The Collaborative Network Relationship Analysis (CNRA) is based on the SNA and focuses on the relations between independent organizations collaborating in networks (Jagdev & Thoben, 2001) and forming a Collaborative Networked Organization (CNO) (Camarinha-Matos et al, 2009). It analyzes the type and intensity of interactions between those organizations (Eschenbächer & Thoben, 2009; Eschenbächer et al, 2009). The CNRA still lacks the inclusion of dynamic changes in such relationships caused by continuous changes in network constellations over time.

Dynamic Network Analysis (DNA) has been developed to overcome the limitation of considering static network models and includes a dynamic component (Carley et al, 2007). The heart of this approach is the so-called meta-matrix which allows defining multi-modal, multi-plex dynamic networks supporting various properties and measures connected to nodes, edges and graphs.

This paper presents an extension to the CNRA approach based on DNA to include mechanisms allowing the modelling of dynamic aspects of Collaborative Networked Organizations. The focus of the modelling is still on different types of interactions between organisations in a network but includes their evolution. This allows a more

comprehensive and prospective evaluation of the performance and the design of the network. The dynamic extension of CNRA is called the DCNRA. A short example explains the application of the DCNRA.

As the traditional SNA approaches often use graph theory terminology and notation (Wasserman & Faust, 2007), this style is also used within this paper.

2 Current State-of-the-Art

This section provides a short overview on what has been done in Collaborative Network Relationship Analysis and introduces the MetaMatrix approach which is the basis for Dynamic Network Analysis.

2.1 Summary of the CNRA

The Collaborative Network Relationship Analysis (CNRA) has first been introduced by Eschenbächer (Eschenbächer et al, 2009). The general idea behind this approach is the quantification of relations between organizations engaged in a Collaborative Networked Organisation as defined by the ECOLEAD project (Camarinha-Matos et al, 2008). The quantification of inter-organizational relations allows the calculation of specific indicators measuring among others the collaboration intensity in the network.

The CNRA knows just one entity class (node type) representing organizations, but introduces several relation (edge) types representing different categories of interactions (see Fig. 1). For each existing edge an intensity is calculated based on estimations for a set of variables connected to that category. By simply summing all intensities of all categories between two given nodes, the collaboration intensity between those organisations is calculated.

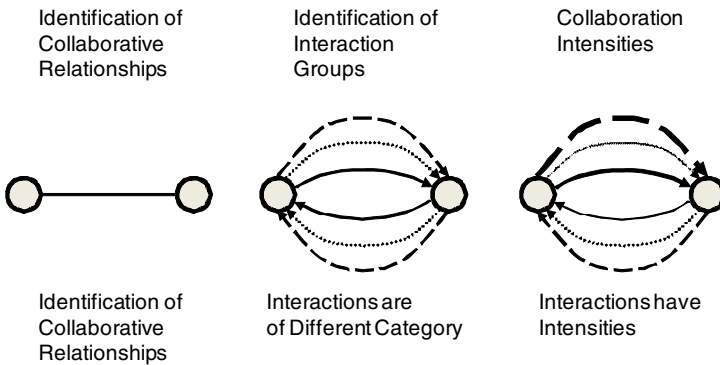


Fig. 1. Concept of Collaborative Network Relationship Analysis

The first application area of the CNRA were innovation related projects executed in a consortium (Eschenbächer et al, 2009). The first approach to dynamics with a project management example was introduced by Eschenbächer, Duin and Thoben (2009) by introducing a start and end point in time to collaborative interactions. This

allows the calculation of how strongly specific partners are involved in the execution of tasks, but still lacks the differentiation of interaction types and only has one visualisation of the Collaborative Network covering the time span from beginning to end. There is no dynamic change of the underlying graph representing the Collaborative Network.

2.2 The MetaMatrix Approach

The MetaMatrix emerged from the work of the Center for Computational Analysis of Social and Organizational Systems (CASOS) located at the Carnegie Mellon University of Pittsburgh (USA). The objective behind the MetaMatrix is to introduce dynamics to Social Network Analysis (SNA) approaches (Carley at al, 2007).

The MetaMatrix is a multi-mode, multi-plex approach to organizational design and serves as ontology. The main entities classes are agents, knowledge, resource, task, organization and location. Nodes belonging to any two entity classes and their relations (edges) form a network like a social network or a knowledge network (knowledge by agents). Furthermore, there can exist several types of relations between nodes of two given entity classes, e.g. “is related to” or “receives instructions from” for nodes belonging to the agent entity class.

This approach seems to be perfectly suited to extend the CNRA with dynamic capabilities.

3 Extending the CNRA to the DCNRA

The basic ideas of the MetaMatrix approach are taken to extend the current CNRA approach. This includes an extension of the type of nodes, an analysis of the type of edges and the addition of properties allowing dynamic changes in the models. As this paper introduces the basic conceptual ideas behind DCNRA no formal notation is introduced which has to be done in future work.

3.1 Extending Node Types

Nodes which do only represent organizations in the CNRA approach are extended to represent other entity classes necessary to model the involvement of organizations in tasks providing necessary knowledge and/or resources. This leads to the definition of the following entity classes (Carley at al, 2007):

- Class 1 – **Agent**: A node of this type represents a business organisation (enterprise or company), a research organisation, a public authority or a natural person. For diagramming collaborative networks we use a circle to represent nodes of this type.
- Class 2 – **Task**: A task is another type of a node representing an operational job where one or more agents are involved to complete the task. A task is a time consuming operation which normally provides some kind of result at its completion time. For diagramming purposes we use a parallelogram representing a task node.

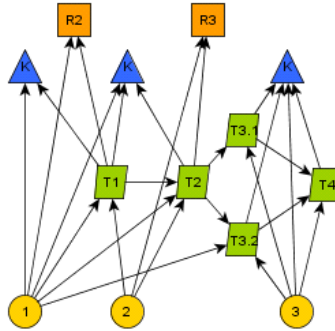


Fig. 2. Modelling example with this approach

- Class 3 – **Resource**: A resource is any kind of machine, supporting material, working power, or other resource like energy and water. In diagrams a square is used to represent a resource node.
- Class 4 – **Knowledge**: Knowledge can be any information stored in the brains of employees or in computers as well as competences of staff. For diagramming collaborative networks we use a triangle to represent nodes of this type.

Note: in difference to the MetaMatrix there is no differentiation between agents and organizations. These two classes are represented here by just one class: the agent. Another class – the location – is also missing, but can easily be included if necessary.

It makes no sense to allow edges between all of the entity classes (node types). On the other hand there may be more than one relational type between two given entity classes. A first approach to the modelling of relations is provided in Table 1. Most of

Table 1. Possible sets of relations and their interpretation

	Agent	Task	Resource	Knowledge
Agent	hierarchical relation (belongs to)	is involved in task	has the resource	has the knowledge
Task	\emptyset	order relation (follows); hierarchical relation (belongs to)	needs the resource	needs the knowledge
Resource	\emptyset	\emptyset	hierarchical relation (belongs to)	\emptyset
Knowledge	\emptyset	\emptyset	\emptyset	hierarchical relation (belongs to)

the possible relations are intuitive and do not need further explanation. The reverse relations (e.g. Task \rightarrow Agent, meaning that a task is performed by one or more agents) might be included for better graph traversal support, but is not necessarily needed as this information can also be retrieved by generating the set of all agents executing this task.

A set of interconnected tasks with relations of class “is followed by” can be considered as a project. The corresponding consortium (forming a Virtual Organisation) is the set of agents which have a “is involved in” relation to the tasks setting up the project. Fig. 2 shows an example of a CNO modelled with this approach. There are three organisations involved in the collaborative execution of a couple of tasks. For task execution some kind of resources and knowledge is needed and provided by the involved organisations.

All entities and relations can have additional attributes. Especially the relation Agent \rightarrow Task should have an entity story the intensity of involvement of the agent in that task. A first approach could be store the percentage of the total working power of that agent put at the corresponding task.

3.2 Adding Dynamic Properties

Each of the entity classes and their corresponding relations can have dynamical properties as shown in Table 2.

The relation class Task \rightarrow Task can be twofold, i.e. describing that one task follows another one or that the task is a sub-task of another one. Both relation classes don't have direct time related attributes, but can be used for consistency checks, e.g. whether a follow-up task really starts after the predecessor finishes.

Table 2. Time related properties of relations

Entity Class	Time Related Properties
Agent	begin (birth), end (dead)
Task	planned start and end, real start and end
Resource	begin (e.g. date of acquisition), end (end-of-life or selling), availability, usage
Knowledge	creation date, (outdating date)
Relation Type	Time Related Properties
Agent \rightarrow Agent	begin, end
Agent \rightarrow Task	beginning and end of involvement (planned and real)
Agent \rightarrow Resource	begin, end
Agent \rightarrow Knowledge	begin, end
Task \rightarrow Task	-
Task \rightarrow Resource	begin, end
Task \rightarrow Knowledge	begin, end
Resource \rightarrow Resource	-
Knowledge \rightarrow Knowledge	-

Table 2 summarises time related attributes given to entity and relational classes. All of them have some kind of a beginning (start or birth) and an end-date of their existence. With the help of the definition of these times related attributes a kind of dynamics can be introduced. For a given point in time t only those entities are existent where t is between the beginning and the end of the existence. Formally, all the other entities do exist as well but only virtually. They are not visible for that moment t . When moving the along a timeline the shape of the considered graph is changing depending on the life spans of the single nodes and edges. A few examples are shown in the following chapter.

Several time dependent indicators can be calculated, stored and dynamically updated. Examples are:

- Collaboration intensity of an agent for a given time span.
- Cumulative collaboration intensity for a set of tasks.
- Needed collaboration intensity for the execution of a set of coherent tasks.
- Total planned collaboration intensity for a Virtual Organisation
- Total real collaboration intensity for a Virtual Organisation
- etc.

Time related attributes for nodes of the task class are differentiated in planned and real beginning and end. This allows executing a simulation of the task models by making assumptions on the real start and end for the considered tasks. One option could be to define a probability distribution function which could be used by a simulation algorithm to determine the “real” start and end by a random generator.

The corresponding changes in the graph over time can be performed by graph grammars or graph transformation systems.

4 Illustrative Example

This chapter introduces and explains some illustrative examples covering basic collaborative interactions such as information exchange, cooperative planning, and collaborative execution of tasks. Similar situations are found e.g. in the execution in European research projects. Concerning the modelling of the research information there is the Common European Research Information Format (CERIF) (Jörg, 2008) which might be used in conjunction with the DCNRA approach.

4.1 Information Exchanges

Information exchange between two partners in the network is modelled using a task which changes the relation of another partner to a specific knowledge entity.

The situation is described by Fig. 3. At a given point in time t_0 there are two partners, one of them (partner 1) having specific knowledge. Both of them are involved in a knowledge transfer task at a later point in time, t_1 . During this task another relation to the same knowledge item is established originating at the other partner (partner 2). Later at t_2 the task has finished, and the “birth date” for partner 2 having specific knowledge is during the execution time span (or latest at the real end of that time span) of the corresponding knowledge transfer task.

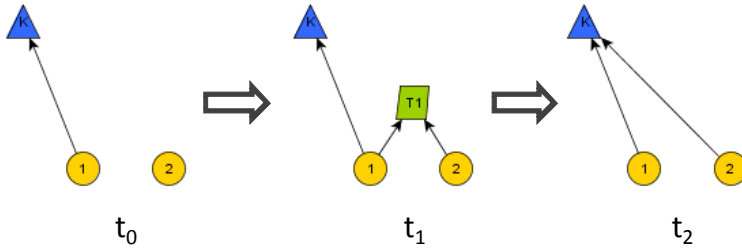


Fig. 3. Example: Knowledge transfer between two organisations

4.2 Collaborative Execution

The collaborative execution of the modelled project example shown in Fig. 2 is presented in Fig. 4. Time t_0 only task T1 is active. The other tasks and their relations are not visible. At time t_1 task T1 has been finished and disappeared. Task T2 is now active and visible. In the next considered time step t_2 task T2 is also finished and tasks T3.1 and T3.2 are executed in parallel. During the first two tasks partners 1 and 2 are involved while in executing T3.1 and T3.2 partners 1 and 3 are involved.

5 Conclusions

The presented concept has the potential to enhance the Collaborative Network Relationship Analysis with a dynamic component. This approach allows the visualisation and simulation of dynamic changes while a Collaborative Network is operating and developing. Additionally, it supports some kind of consistency checks, e.g. whether the involved agents do have the necessary capacities in terms of knowledge and resources. The definition of indicators allows the dynamic calculation of performance and collaboration intensities of involved partners.

Planners of Collaborative Networked Organisations are supported with a tool allowing the beforehand simulation and validation of that CNO. Several scenarios having changed or balanced collaboration intensities can be generated and compared. This allows also an assessment of the risks associated to each of the scenarios.

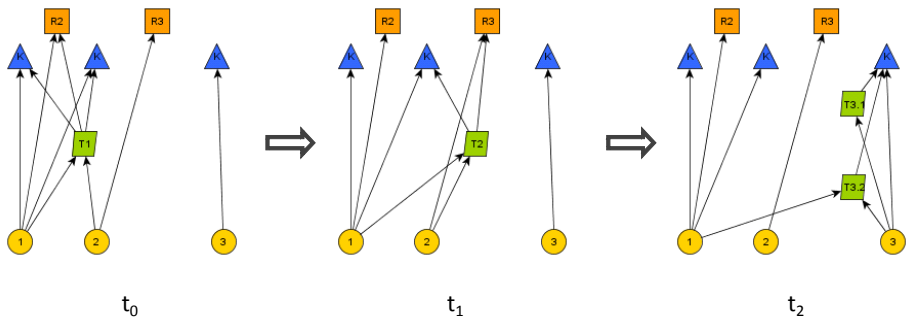


Fig. 4. Example: Collaborative execution of tasks

However, this approach is still in its conceptual phase. First ideas are presented here and further development of the formalism (including indicators and their formulas) is still needed. An example case study needs to be executed in a master project to show that static SNA results are not helpful in a complex and dynamic world. Also an application in real world contexts is still missing but necessary to prove its usefulness.

Acknowledgements. This work has been partly funded by the European Commission through ICT Project COIN: Collaboration and Interoperability for networked enterprises (No. ICT-2008-216256). The authors wish to acknowledge the Commission for their support. We also wish to acknowledge our gratitude to all COIN project partners for their contribution.

References

1. Camarinha-Matos, L.M., Afsarmanesh, H., Galeano, N., Molina, A.: Collaborative Networked Organizations - Concepts and Practice in Manufacturing Enterprises. *Computers and Industrial Engineering* 57, 46–60 (2009)
2. Camarinha-Matos, L.M., Afsarmanesh, H., Ollus, M.: *Methods and Tools for Collaborative Networked Organizations*. Springer, New York (2008)
3. Carley, K.M., Diesner, J., Reminga, J., Tsvetovat, M.: Toward an Interoperable Dynamic Network Analysis Toolkit. *Decision Support Systems* 43, 1324–1347 (2007)
4. Jagdev, H.S., Thoben, K.-D.: Anatomy of Enterprise Collaborations. *Production Planning and Control* 12, 437–451 (2001)
5. Jörg, B.: CERIF: The Common European Information Format Model. Insight into the CERIF 2008 Release. In: 9th International Conference on Current Research Information Systems (CRIS 2008), Maribor, Slovenia, June 5–7, pp. 5–7 (2008)
6. Eschenbächer, J., Duin, H., Thoben, K.-D.: Planning Distributed Innovation Processes in Virtual Organisations by Applying the Collaborative Network Relationship Analysis. In: Thoben, K.-D., Pawar, K.S., Katzy, B., Bierwolf, R. (eds.) *ICE 2009 - Proceedings of the 15th International Conference on Concurrent Enterprising: ICE 2009 - Collaborative Innovation: Emerging Technologies, Environments and Communities*, Leiden, The Netherlands, June 22–24 (2009)
7. Eschenbächer, J., Seifert, M., Thoben, K.-D.: Managing Distributed Innovation Processes in Virtual Organizations by Applying the Collaborative Network Relationship Analysis. In: Camarinha-Matos, L.M., Paraskakis, I., Afsarmanesh, H. (eds.) *Leveraging Knowledge for Innovation in Collaborative Networks*, pp. 13–21. Springer, New York (2009)
8. Wasserman, S., Faust, K.: *Social Network Analysis. Methods and Applications*. Cambridge University Press, New York (2007)