

W. Mulder,
LogicaCMG Netherlands, University of Amsterdam, wico.mulder@logiacmg.com
P. H. H. Rongen,
IBM Netherlands, erik@nl.ibm.com
G. R. Meijer,
University of Amsterdam, LogicaCMG Netherlands, Geleyn@science.uva.nl

As the interest in Collaborative Networks grows, and the number of organizations participating in a Virtual Breeding Environment increases, tools will be needed to support the creation of Collaborative Network Organizations. An important driver for this support is that in modern commercial environments a rapid response to changes in the market is essential to remain competitive. We are working on a domain-ontology and a model that supports the composition of CNOs using semantic capability descriptions. We apply this model in the field of software development, where multiple expertise teams form dynamic task-forces, called Squads, in order to build and manage computer applications. In this paper, we focus on the development of the domain-ontology and the elements of the composition model. We are working on a prototype that supports the creation of Squads by means of capability matching.

1. INTRODUCTION

The trend in the ICT market in shifting revenues from the sales of products towards the provision of on-demand services continues. In order to remain competitive in an agile market, organizations need to be able to respond rapidly to changes. As a result, collaborative strategies and flexible forms of cooperation between organizations emerge.

The discipline of collaborative networks (CNs) is defined to focus on the structure, behavior, and evolving dynamics of networks of autonomous entities that collaborate to better achieve common or compatible goals [Camarinha]. Variants of CNs, such as Virtual Enterprises (VEs), Virtual Organizations (VOs) and Virtual Breeding Environments (VBEs) form part of our research.

Today, the concept of VOs is familiar, and related environments, such as VBEs, are getting significant momentum. A VBE represents an association (also known as cluster or pool) of organizations that have the potential and the intention to cooperate with each other through the establishment of a long-term cooperation agreement and interoperable infrastructure [Camarinha]. In case a business opportunity is identified by one member, a subset of these organizations can be selected to form a CNO.

If the number of organizations in a VBE is large, the selection and formation needs to be supported by a semi-automatic system. Such a system might relieve an initiator inside a VBE from unnecessary complexities of specifying the organizations to form the CNO that has to fulfill the job. The objective of such a

system is to match a particular business case, for example a customer problem, with the capabilities of the VBE-members. We are developing a model and a prototype to support this kind of matching. It is here where we enter the field of domain ontologies, semantics, and reasoning.

Ontologies are a popular research topic in various communities such as knowledge engineering, natural language processing, cooperative information systems and knowledge management. They are used to share a common understanding of a domain that can be communicated across people and computers. They are also used in the field of team matching and competence management [Posea,Hefke].

In previous work [Mulder] we described the concepts of collaborative organizations that develop and maintain software. We defined the term *Squads* as mobile teams that operate in the field of software development and application management. Here, in this paper, we discuss the ontology-based composition support, a Squad is described as a task-force consisting of one or more expertise teams.

While the size of expertise teams varies from a single person to teams that can be identified as sub-organizations within a large company, they can be grouped in Squads. Note that the model can be applied to expertise teams from different organizations, but can also be applied to expertise teams within one single organization. A mechanistic view on expertise teams models them in the form of a processing unit with an interface containing input and output parameters. A Squad can be seen as a chain or network of expertise teams each having an input and output interface.

Figure 1 shows a VBE containing a set of expertise teams. The teams are willing and have potential to participate in Squads. They have registered themselves as a member of the VBE. In case a customer provides one of them with a problem, the process of selection and matching takes place. This will be followed by the formation of a Squad whose mission is to provide a solution for this customer.

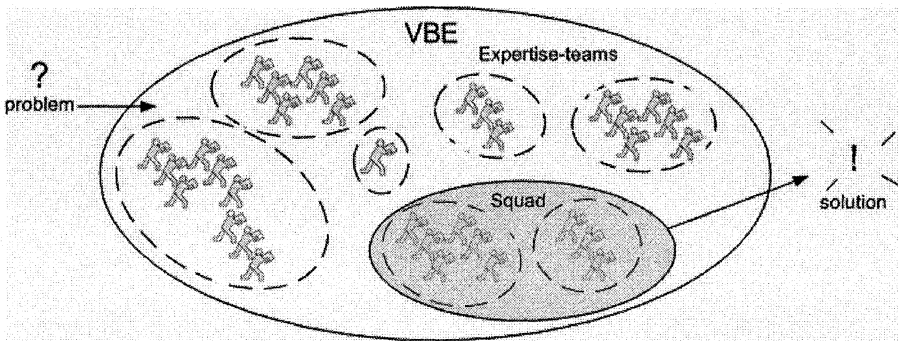


Figure 1 - A VBE containing expertise teams.

The field of automatic composition of small entities to provide end-to-end services [Rabelo] is not new. In [Kvaloy] the ontology based composition and selection of semantic web-services is described. Individual web-services are grouped together in an application flow to perform the requested task. In the area of semantic web-services, ontologies are being used intensively [owl-s]. The ideas of expertise team matching show remarkable similarities with this area. Also in the field of agent technology strategies exist to automate the process of negotiation [Chen, Dignum, Norman]. We adopt the results from these areas in the new area of collaborative networks. Driven by the need for tools that support CNO-management, we work on ontology-based support for the creation of Squads.

Figure 2, contains four phases as part of a Squad-life-cycle-model. The matching phase starts with the analysis of an incoming customer problem and is followed by the selection and optimization of expertise teams that have subscribed themselves as members of the VBE. In case a match between the expertise team capabilities in combination with the problem is found, the phase of physical Squad-formation might begin. This paper focuses on the support for selecting candidate squads, denoted as the matching phase.

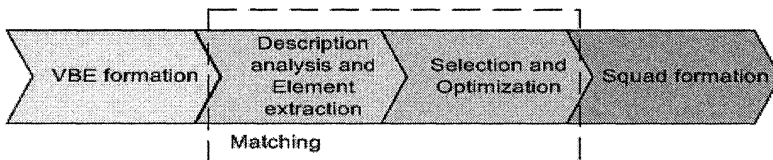


Figure 2 - The matching phase

The rest of the paper is organized as follows: section 2 describes the structure of the expertise team capability descriptions, which forms the basis for our domain ontology. In section 3 we describe the model for the automated matching process and mention our prototype of the matching tool. Section 4 contains results, section 5 contains discussion and future work, and section 6 ends with the conclusions.

2. A DOMAIN-ONTOLOGY FOR SQUADS

In general, collaborative parties have to use common terminology to express their role and contracts. This terminology, represented by a domain-ontology, is necessary for each type of interaction between organizations. In our approach the organizations are reflected by expertise teams, and the collaborative environment is a software development VBE. By describing the capabilities of individual expertise teams we are developing a domain-ontology that reflects the knowledge and expertise of the VBE. Motivated by ideas of web ontologies [owl-s], we distinguish three types of descriptions:

- The profile: the needs from others and the deliverables of the team.
- The model: description of how an expertise team performs its tasks.

- The *grounding*: describes the context of the team in which it operates

The *profile* describes capabilities, conditions and prerequisites. On one hand it tells what the expertise team can deliver, on the other hand it expresses needs from others in order to deliver its service.

The *model* describes how the expertise team works, and the steps involved while carrying out its service. The description can be used to perform a more in-depth-analysis, or to enhance capability descriptions.

The *grounding* describes in what situations the team is normally involved, and how they can be reached during their operation.

An example is a Java web-development team subscribing itself as a member of the VBE. The profile may describe that the team provides web interfaces for existing systems, based on use cases and technical designs. The model describes that they use RUP as a development methodology, the types of common frameworks they use and their development environment tools. The grounding describes that the team is usually involved in development projects for customers in financial markets and that they can be accessed by their organization's local management. Other examples are architect-teams, C++-teams, database-teams and teams of functional analysts.

In our model the capabilities of expertise teams are written in the language that is represented by our domain ontology. We use this ontology in an environment that supports the creation of Squads by matching customer requests with expertise team capabilities. The relations between the entities in the model allow us to analyze the capability descriptions and apply a form of description logic reasoning. We believe that the three kinds of descriptions not only lead to proper matching possibilities, but also reflect our domain-ontology in a good sense. The focus on capabilities and the development of the ontology will go hand in hand.

3. CAPABILITY MATCHING

We distinguish two important aspects of matching; capability analysis and configuration optimization.

The first aspect, capability analysis, covers the analysis of problem descriptions and expertise team capabilities. As shown in Figure 3, the domain-ontology is used to extract key-elements from a particular problem description. The same is done for the expertise team capability descriptions. The elements (denoted with K's) that occur as classes in the ontology are called *direct* elements. These elements are used to infer other domain elements by means of relationships and attributes described by the ontology. For this, an inference engine can be used. Comparing the elements found for the problem description with those derived from the expertise team capabilities, results in a similarity-score that represents the level of matching.

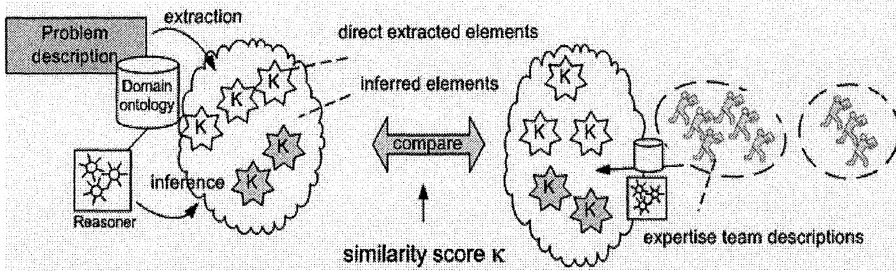


Figure 3 - Matching using (inferred) domain keywords

The second aspect of matching contains preferences and optimization rules in order to combine expertise teams into candidate squads. In our analysis we work towards a utility function that can be used to find the most suitable Squad.

The capabilities of a squad are reflected by the capabilities of its constituting expertise teams. Some capabilities depend on each other, for example the ability to program Java applications is necessary for the ability to create J2EE applications. Other capabilities are independent, which means that having one capability does not imply the presence of another. We represent the requested and available capabilities in a multi-dimensional space. The individual, independent capabilities of the expertise teams in the VBE span a multi-dimensional space schematically shown in Figure 4. Part (a) shows the volume within that space containing the required capabilities to build a solution. This volume, represented by the dashed line, will hereafter be called the requirements space. The goal is to combine expertise teams to fill the requirement space in the best possible way.

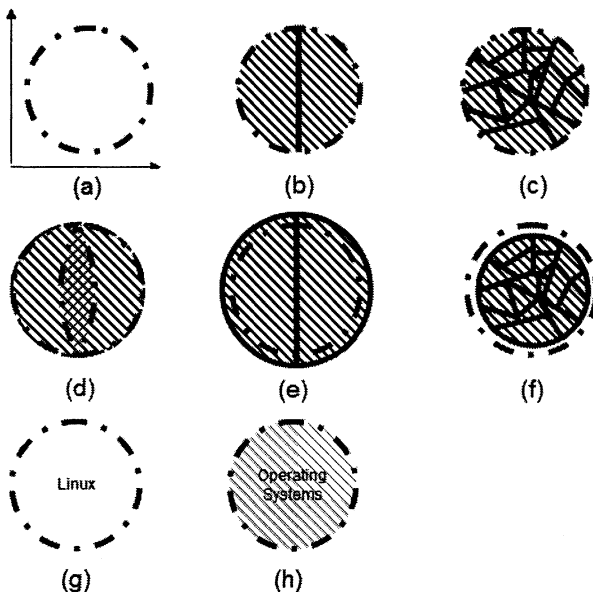


Figure 4 - Capability matching and squad formation.

Figure (b) shows the situation where two expertise teams are combined into a candidate squad. The combined capabilities, represented by the dashed area of these two expertise teams exactly match the requirements space. In (c), the requirements space is also exactly matched, but now the Squad consists of many small expertise teams. In (d), the requirements space is again exactly matched by a squad, but now the expertise teams of the candidate squad have an overlap in their expertise. In situation (e) the combined capabilities of the squad are partly outside of the requirements space, meaning the squad has capabilities that are not asked for, and are possibly irrelevant for the task at hand. Figure (f) shows the situation where the squad does not match the total requirements space. Figures (g) and (h) represent the situation where a requirement (e.g. Linux skills) is not matched, but expertise of a more general concept in the ontology is present (e.g. Operating systems).

The goal of our system is to define the best squad for a given task as specified by the requirements space. In many cases, there will be various alternatives to form Squads out of expertise teams that match the requirement space.

In our current work we focus on three kinds of considerations:

- Many expertise teams versus a few expertise teams
- Overlap of relevant capabilities of the expertise teams in the Squad
- Capability coverage of the requirement space

Figure 4 (b) and (c) show the requirements that are met by a few or many expertise teams. The disadvantage of having a Squad consisting of many expertise teams could be that more agreements between the teams have to be made. It is likely that it will take a lot of effort to come to these agreements, and also that during the tasks performed by the Squad, communication between these expertise teams will consume a considerable amount of time. On the other hand, having more expertise teams result in less supplier-dependent solutions.

A different consideration is about expertise teams having overlapping capabilities (d). Although this might be an overkill of required skills, Squads are more able to compete and communicate, improving the quality of the solution.

The third consideration describes the capabilities required versus those that can be provided. Figure (e) shows the case that the expertise teams have more capabilities than required. This might be more expensive, since the expertise teams have experiences in fields which are not required for the task at hand. However, if those capabilities are related to the ones in the requirements space, this can also be an advantage, because the Squad can deal with unexpected problems that are related to the problem it is being created for.

Opposite to this situation is the case in which not all capabilities are available in the VBE to fulfill the task (f). Here, none of the possible squads can fulfill the requested tasks, which brings us to the third kind of coverage-problems: only capabilities of a more general concept in the ontology are available (g) and (h).

The above discussion tells that there is no unique way of describing these features in terms of positive or negative influence on the utility of a Squad to achieve its goal. This means that in most cases preferences are needed to find the optimal Squad for a particular task. Also, personal preferences about the teams might be of influence.

In our model the optimization will be achieved by maximizing a particular utility function with parameters representing the features of a Squad as discussed above:

- Number of expertise teams in the Squad
- Overlap size of capabilities of the expertise teams forming the Squad
- Capability coverage of requirements and capabilities

4. RESULTS

We are implementing the model described above in a prototype environment called SqoMMe, which stands for Squad ontology-based Matching and Managing environment. While identifying candidate expertise teams and making suggestions about optimal Squad configurations, the environment interacts with the user.

We are in the process of developing our domain-ontology using the PROTÉGÉ [protege] environment. The first version consisted of two groups of classes. One group is process-related, containing expertise-team, capabilities, task and mission. The other group is technique-related, containing programming languages, software components and their relationships. In parallel with these ontology-explorations, we are working on implementations of matching mechanisms in Java, mainly to study the effects of various approaches. A third activity, which is also in early-stage, is to find a proper way for doing inference reasoning.

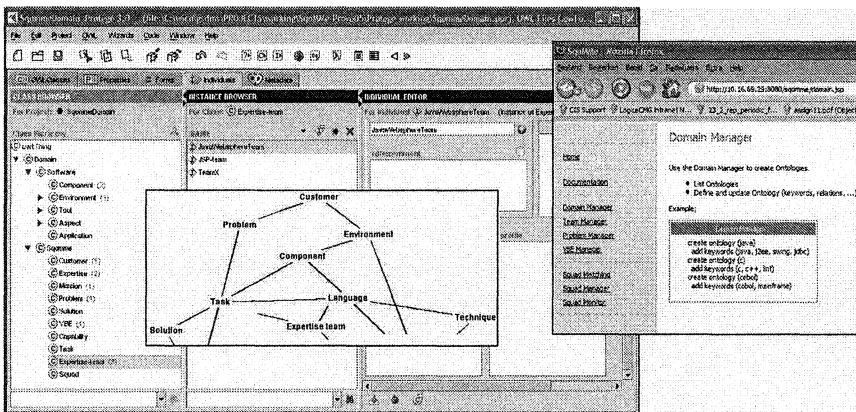


Figure 5 - Screenshots of the protégé and SqoMMe environment

5. DISCUSSION

At the moment the model does not involve availability issues and service level agreements (SLAs) of the individual expertise teams. In our current work we prioritize to focus on the matching of capabilities. For the same reason the aspect of costs is left out. We recognize that it might have a strong influence in the matching phase, although it might also have its main relevance in the formation phase. In future work, we want to include both aspects in our model, together with the lessons learned from testing the environment in our own company.

We also study whether the ontology used for matching can support the monitoring and management of Squads as well. Therefore we will investigate the use of the same domain-ontology to enable the automation of Squad performance measurement, which contributes to the support for monitoring and managing the Squads during their operation.

6. CONCLUSION

In this paper we presented a model to support the creation of Squads, dynamic task-forces that operate in the field of software development.

We discussed the way of describing the capabilities expertise teams, and the use of a domain-ontology for matching these capabilities with problem descriptions. We are applying the model in a prototype and study the use of a utility function to find the best suitable configurations of Squads. Our research on this subject is in early stage, but is expected to be of important value in the creation of CNOs.

7. ACKNOWLEDGEMENTS

This work is supported by the Ecolead project.¹

We would like to thank Ron de Kruijff and Tim van Soest for their suggestions, and René Kuipers for all the fruitful discussions and technical efforts.

8. REFERENCES

1. Camarinha-Matos, L.M., Afsarmanesh H, The emerging discipline of collaborative networks, In *Virtual Enterprises and Collaborative Networks*, (L. M. Camarinha-Matos, Editor), Kluwer Academic Publishers, ISBN 1-4020-8138-3, IFIP Vol. 149, Aug 2004
2. Chen, Y et al, 1999, A negotiation-based multi-agent system for supply chain management, *Proceedings of the Agents'99 Workshop: Agent-Based Decision-Support for Managing the Internet-Enabled Supply-Chain*, Seattle, WA, 15-20.
3. Dignum, M.V., Dignum, F.P.M., & Meyer, J-J.Ch. 2004. An Agent-Mediated Approach to the Support of Knowledge Sharing in Organizations. UU-CS-2004 (Ext. r. no. 019). Utrecht: UU-CS-Technical Report
4. Hefke M, Stojanovic L., An ontology-based approach for competence bundling and composition of ad-hoc teams in an organization, in K. Tochtermann and H. Maurer, *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW'04)*, Journal of Universal Computer Science (J.UCS), Graz, Austria, June 2004.
5. Kvaloy, T.A., R.H.H. Rongen, Automatic Composition and Selection of Semantic Web Services, *Proceedings EGC, Amsterdam, 2005*
6. Norman, T.J. et al., Agent-based formation of virtual organizations, *Knowledge-Based systems 17 (2004) 103-111*
7. Mulder W., Meijer G.R., SQUADS: Software Development and Maintenance on the Grid by means of Mobile Virtual Organizations using Adaptive Information Services, *Virtual Enterprise and Collaborative Networks, Proceedings PROVE04, WCC 2004*, ISBN 1-4020-8138-3, p. 379-386
8. (Owl-s) <http://www.daml.org/services/owl-s/1.0/>
9. Posea, V., Harzallah M., Building an Ontology of Competencies, in *Workshop on Ontology and Enterprise Modelling: Ingredients for Interoperability Proceedings*, Vienna, Australia, Dec 2004
10. (Protégé) <http://protege.stanford.edu>
11. Rabelo R. J., Baldo F., Tramontin R. Jr., Pereira-Klen A., Klen E.R., Smart Configuration of dynamic, Virtual enterprises, *Proceedings PROVE04, WCC 2004*, ISBN 1-4020-8138-3, p. 193-203

¹ www.ecolead.org