A Hierarchical Super Peer Network for Distributed Software Development[1]

Ludger Bischofs[1], Wilhelm Hasselbring[2]

[1] OFFIS, Escherweg 2, 26121 Oldenburg, Germany, ludger.bischofs@offis.de

[2] University of Oldenburg, FK-2, Software Engineering Group, 26111 Oldenburg, Germany, hasselbring@informatik.uni-oldenburg.de

Abstract. *The approach presented in this paper is the construction of a peer-to-peer based network for distributed software development, which maps the structures of open source projects in a natural way. Distributed teams, in particular for open source projects, can be considered as peer-to-peer systems themselves. A hierarchical super peer network is being designed to support the formation of new developer groups and projects and to allow a flexible self-organisation of the organisational units and their relationships. Distributed resources like version control systems as well as local devices can be integrated into the network. Basic synchronous and asynchronous communication with the possibility to use external CSCW systems for intensive coordination and cooperation enhance the network.*

1. Introduction

The contribution of this paper is the construction of a P2P (peer-to-peer) based network for distributed software development, which maps the structures of open source projects in a natural way. Distributed teams, in particular for open source projects, can be considered as P2P systems themselves. The network support already begins by the formation of new developer groups and projects and allows a flexible self-organisation of the organisational units and their relationships. Distributed (maybe specialised) developers and developer groups will be supported without narrowing down their software engineering process. Thus the developers and developer groups are granted a widely autonomy to improve their group dynamics.

Distributed resources like version control systems (e.g. CVS or local devices) can be integrated into the network. Basic synchronous and asynchronous communication with the possibility to integrate CSCW (Computer Supported Cooperative Work) systems like mailing lists or forums for intensive coordination and cooperation enhance the platform.

This paper is organised as follows. Section 2 describes specifics of OSS (open source software) development which have to be considered in the design of a comprehensive network for distributed software development. In section 3, peer types are identified representing the four central organisational units that are used to map the structures of OSS development projects into a P2P architecture. The relationships of these organisational units respectively peer types are described in section 4. The integration of resources like version control systems or CSCW systems are demonstrated in section 5. Section 6 shows how the peers can be organised in a hierarchical super peer network and how an appropriate lookup service will look like. In section 7, a possible architecture of a P2P application is depicted which fulfils the requirements of such a network. After presenting related work we conclude and outline future work.

[1] This work is supported by the EU project DELOS (Network of Excellence), contract nr. 507618

2. Open Source Software Development

OSS development often begins uncoordinated and evolutionary. Different projects are often managed in radically different ways, because there is no single OSS development process. The OSS process generally involves (or has the potential to involve) large, globally distributed communities of developers collaborating primarily through the internet. These developers tend to work in parallel, with different individuals/groups working on different aspects of the system simultaneously. Many projects live from developers who join it voluntary, so that they are not allocated to a project by a supervisor. In fact the developers pool together worldwide to cooperate because of a common interest. Many small to medium OSS projects are coordinated by a single individual or by small, unofficial groups. However, most of the major OSS projects are coordinated by official or quasi-official organisations.

The OSS development life cycle is quite different from the traditional software development life cycle, which typically includes planning, analysis, design and implementation. In OSS projects the project founder largely conducts the planning, analysis and design phases. That's why they are not part of the general OSS development life cycle, which is located primarily within the implementation phase of the traditional software development life cycle [4].

3. Peer Types and Organisational Units

P2P is characterised as the opposite of client/server architectures. The most distinctive difference is that in a P2P network the nodes act as server and client at the same time. The participants share a part of their own hardware resources like processing power, storage capacity or network link capacity. The peers are accessible by other peers directly without passing intermediary entities [11].

This paper introduces a P2P based OSS development network which connects developers so that they are able to build groups, start projects and collaborate flexible. Those projects can grow evolutionary. The four major groups involved in OSS are developer communities, user communities, commercial and non-commercial organisations [4]. Many OSS users are also developers and many developers are also users. In our approach both communities are represented by one type of peer: the *single peer*. In distributed software development projects, developers may be distributed worldwide and can group together to reach a common goal. Therefore *group peers* are helpful for group management. Nevertheless, developers also can work alone or together without a group peer. Groups can participate in more than one project, e.g. if it develops a component, which can be reused in different projects. The introduction of *project peers* is needed to offer project management functionality. Organisations often assert highest level of control over the direction and future of projects. Therefore commercial and non-commercial organisations are represented by *organisation peers*. Organisation peers allow the specific search of resources or groups within them. Also internal areas can be differentiated and protected from the outside world, because only associated organisational units have access rights.

To organise the distributed OSS development we introduce the four central peer types which represent the organisational units:

- A *single peer* typically represents the computer of a software developer or user which offers basic services like registration of users, search functions, data transfer and basic communication. Each peer has the basic functionality of a single peer. The software developer or user is able to search for other developers or for resources and artefacts like documentation, components or executables. Afterwards he can establish a connection to the discovered peer to transfer data or to contact the other developer. It's also possible to grant access to own resources to other users or developers. Additionally, a single peer is able to join a group, a project or an organisation.

- A *group peer* offers services for group management like addition or removal of developers. In so doing developers are organised in groups. A group manager has the ability to associate resources to the group which can be used by the group members. He controls the resources and determines which of them are visible or invisible for other developers.
- A *project peer* offers services for project management. Developers and groups of developers can be organised in projects to reach a common goal. The project manager has the ability to manage common resources for the peers involved in the project like a group manager has for his group.
- An *organisation peer* offers services for organisation management. An organisation can be e.g. an enterprise or institution. The organisation manager has the ability to manage common resources for the organisational units involved in the organisation.

A physical peer can be composed of more than one peer type. For example a computer can act as a project peer and as a group peer simultaneously. By default it also has the abilities of a single peer. Figure 1 depicts a logical view of a possible structure of peers which needs not inevitable be the physical structure of the involved computers. P_2 and G_3 for example can physically be the same computer. In the following section the possible relationships between the peer types are described.

4. Organisational Units and their Relationships

In order to model the relationships between the different peer types the UML notation for aggregation, composition and association is used:

- **Aggregation** and **composition** describe a close cooperation between organisational units respectively peers and also describe their hierarchical order. Typically, to establish an aggregation or composition relationship a peer registers at a superior peer. Thus the superior peer has a special ability to control the registered peers. In figure 1 for example the group peer G_1 has control about the single peers S_3 to S_5. The group peers G_2 and G_4 are connected by a composition relationship. This means that the peer G_4 can not exist without the group peer G_2 whereas in an aggregation relationship the partners of the aggregation can exist without the other one.
- The **association** describes a loosely coupled cooperation with widely autonomous partners. No clear hierarchical structure can be extracted from an association relationship. The groups G_1 and G_2 in figure 1 cooperate. The same is true for the developers within these groups so that access to resources of the respective other group can be granted to them. The developer at single peer S_9 is associated to group peer G_3 which means that he cooperates with the group G_3.

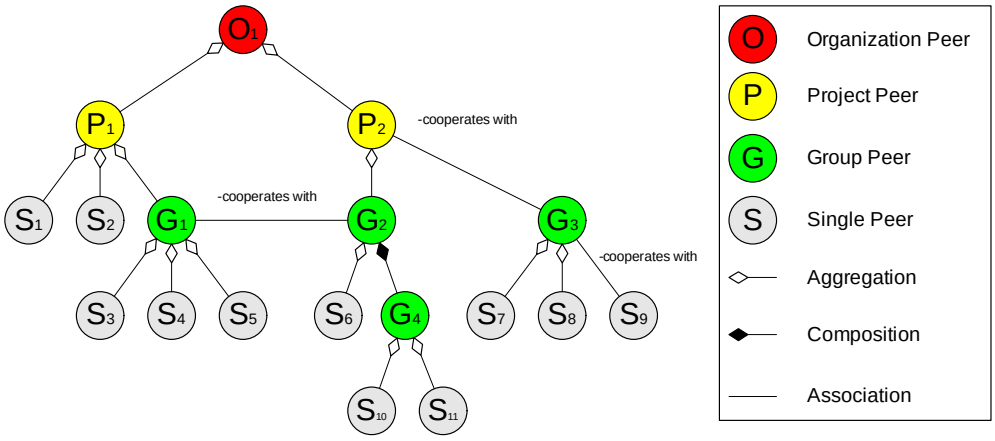

Figure 1: Aggregation, composition and association of peer types

5. Resource Integration

For cooperative software development developer groups typically use a number of tools like version control systems (e.g. CVS or subversion) and CSCW systems. Single developers rather use local devices to store files. In the following tools or storages like the ones listed are understood as resources. A tight cooperation between organisational units is possible by sharing such resources with others. A project peer e.g. typically shares its resources selectively with the involved groups. So the group peers are able to access the resources too. Group peers typically share their resources with single peers, i.e. all developers in a project can use the resources which are connected to the appropriate project peer.

There are two ways to access resources. The developers can exchange credentials and discovery data for resources directly so that they are able to access them without using the P2P network. The alternative is to integrate the resources into the P2P network. Figure 2 shows a solution for this integration problem. The resource peer *R* connects to the resources *A* and *B* in an application specific way. Externally another interface (e.g. in form of a web service) is offered to the rest of the peers. The peers connect to a resource through a special adapter which uses the corresponding external interface of the resource peer. The resource peer either manages access control lists of the peers which have access to the resources or queries another peer, like the project peer *P* in figure 2, whether to grant access to the requesting peer.

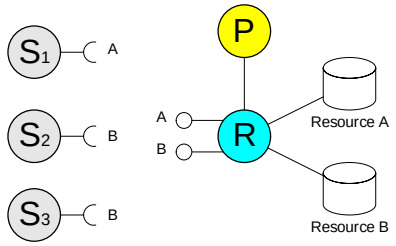

Figure 2: Integration of resources into the Peer-to-Peer network

By using the interfaces of a resource peer other peers can for example access documents which are stored on a CVS server or a local device. Another example for the shared use of resources is the registration of group members for a forum or mailing list carried out by their group peer via a resource peer. So developers can be informed, if important events, like the change of a file, happen.

6. Hierarchical Super Peer Network

The lookup service is a central requirement for P2P systems and assigns and locates resources among peers. Distributed “flat” P2P lookup services are e.g. CAN [9] and Pastry [10]. The approach presented in this paper is the introduction of a hierarchical multi-tier lookup service where peers are organized in disjoint clusters as it is depicted in figure 3. Super peers route the messages along clusters to the destination cluster. Within the clusters the messages move through the hierarchical structure of the peers. The hierarchical peer structures in combination with their super peers form a hierarchical super peer network [2,5]. The super peers hold among other things a common metadata index of available artefacts (e.g. documents, source code) which are distributed over the different organizational units or peer types, respectively. They are able to answer simple queries. Detailed queries additionally pass through the hierarchical structure of the peers. The exchange of the located artefacts takes place directly from peer to peer. Super peer networks have some advantages in comparison to pure P2P networks. They combine the efficiency of the

centralized client-server model with the autonomy, load balancing, and robustness of distributed search. They also take advantage of the heterogeneity of capabilities across peers. The most important benefits of the approach discussed in this paper are scalability and administrative autonomy. A super peer can independently route messages within its cluster. Similarly, organization, project, and group peers can route the messages to subordinated peers using their own strategy so that queries to selected organizational units do not flood the entire network.
The super peers build a common metadata index of the available artefacts within the network. They are connected to each other. Physically the super peers can be any peer who has enough computing power, storage capacity and an adequate network connection. Peers who are registered at a super peer build a cluster as depicted in figure 3.

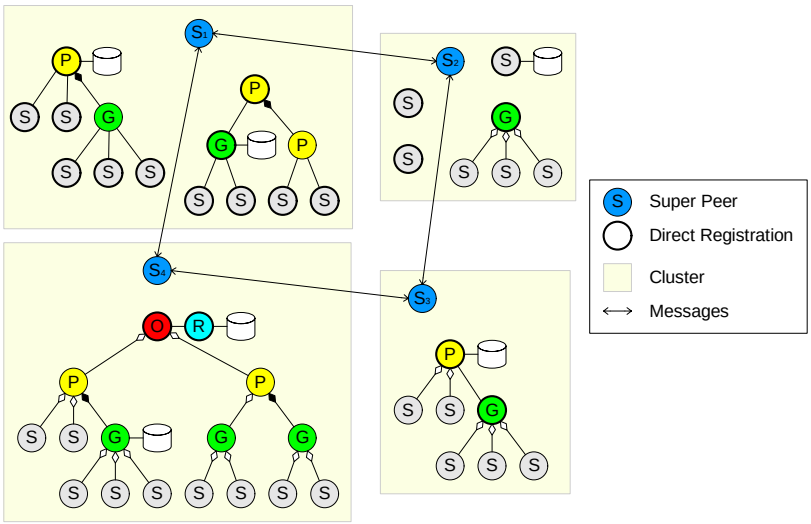

Figure 3: Three-tier Lookup service

There are different ways in which super peers can collect metadata. Peers can register directly at a super peer and send their metadata to it or they can do it indirectly. Typically peers which are only linked by associations to other peers register at the associated peers and send their metadata directly to the super peer whereas peers whose relationship is an aggregation or composition register at the superior peer and send the metadata to it. So the superior peer has extensive knowledge of the subordinated peers. The metadata is send to higher peers in the hierarchical order until it reaches the super peer. The advantage of this approach is that not every peer in a network needs an internet connection to make its metadata available and that the superior peers have control over the metadata which is sent to a super peer. So superior peers have the ability to decide whether metadata of subordinated peers is available or unavailable to superior peers respectively the global index.

Within the metadata index, information on the organisational units like group members, project goals and capabilities of developers are stored. Metadata can also be extracted from resources which are connected to a resource peer. In figure 3 also a simplified view of resource connections is illustrated in which the resource peer is not explicitly visible.

7. Architecture of the Peer-to-Peer Application

Figure 4 shows a possible architecture of a P2P application supporting the hierarchical super peer network. This architecture allows any peer to act as single, group, project, organisation, resource, and super peer at the same time to allow the flexible self-organisation of the organisational units within the hierarchical super peer network. The architecture [7] is divided into a *Communication Module*, application specific modules and a *Graphical User Interface (GUI)*:

- The *Communication Module* represents the physical connection to a network and the connection/communication components of a P2P system. So it deals with the basic P2P communication and organisation that can occur across the P2P network. This could include the passing of messages between peers, the local caching of peer addresses and publishing/discovery mechanisms.
- The *Organisational Unit Manager* handles the relationships between the organisational units controlled by a peer. The manager provides an interface which can be used by other modules to manage the particular organisational units.
- The *Index* offers data search and filter functions. It can gather metadata from the Organisational Unit Manager and the Resource Manager for sending it to a super peer. It also contains the lookup service functionality if the peer acts as a super peer.
- The *Resource Manager* can integrate external resources into the P2P network. It provides interfaces for accessing the resources allowing connections from other peers. Additionally, there is an interface to the Index for generating metadata of resources. The connection to the Organisational Unit Manager is used to query access rights.
- The *Graphical User Interface (GUI)* represents the connection to the user.

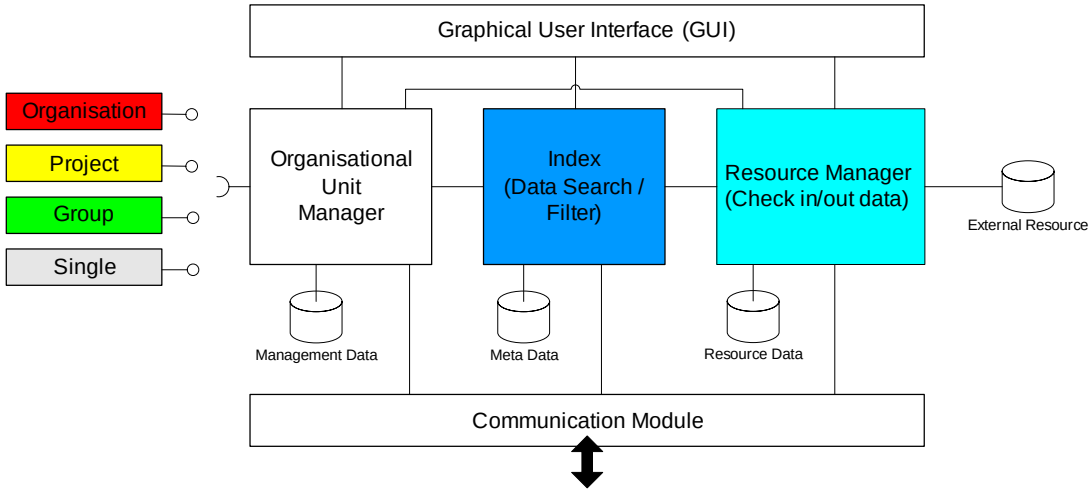

Figure 4: Architecture of the Peer-to-Peer Application

8. Related Work

MOTION (**MO**bile **T**eamwork **I**nfrastructure for **O**rganizations **N**etworking) is a P2P platform for mobile teamwork support. It let employees share business documents, locate expertise and knowledge through distributed searches, access effective subscription/notification mechanisms, and access company's information resources at any time anywhere [6]. The paper [3] presents an approach for a support system based on MILOS (**M**inimally **I**nvasive **L**ong-Term **O**rganizational **S**upport) but altered from a client-server to a P2P application using a P2P framework. GENESIS (**G**eneralized **En**vironm**e**nt for Proces**s** Management **i**n Cooperative **S**oftware Engineering) is an open source distributed platform supporting cooperative software engineering. The requirements have been derived from actual user needs which were collected from some of the largest European software development companies [1].

The designing of super peer networks is described in [12]. The costs and benefits of a new hybrid approach called structured super peers are explored in [8]. It partially distributes lookup information among a dynamically adjusted set of high-capacity super peers to provide constant-time lookup.

In hierarchical P2P systems, peers are organized into groups, and each group has its autonomous intra-group overlay network and lookup service. A general framework and scalable hierarchical overlay management is provided in [5].

9. Conclusions and Future Work

The approach presented in this paper is based on a hierarchical super peer network for distributed software development, which maps the structures of open source projects in a natural way. As we have seen the organisational units which are involved in the process of distributed software development explain a hierarchical P2P system themselves. The designed P2P network supports a flexible self-organisation of the organisational units and their relationships. Further on distributed resources can be integrated into the P2P network. Currently, we work on a prototype fulfilling the requirements of single, group and resource peers. The integration of CVS which is mostly used in open source projects is one of the first major goals.

Also note that this approach of a hierarchical super peer network is not only usable for distributed software development. Other scenarios in different domains like digital libraries [2] are imaginable.

Bibliography

[1] BALLARINI, D., CARDOLI, M., GAETA, M., MANCINI, T., MECELLA, M., RITROVATO, P., SANTUCCI, G., Cooperative Software Development in GENESIS: Requirements, Conceptual Model and Architecture, in: Cooperative Methods and Tools for Distributed Software Processes, 2003.

[2] BISCHOFS, L., HASSELBRING, W., SCHLEGELMILCH, J., STEFFENS, U., A Hierarchical Super Peer Network for Distributed Artifacts, in: Pre-proceedings of the Sixth Thematic Workshop of the EU Network of Excellence DELOS. S. Margherita di Pula (Cagliari), Italy, pp. 105-114, 2004

[3] BOWEN, S., MAURER, F., Designing a Distributed Software Development Support System Using a Peer-to-Peer Architecture, in: 26th Annual International Computer Software and Applications Conference. Oxford, England, 2002.

[4] FELLER, J., FITZGERALD, B., Understanding Open Source Software Development, Addison-Wesley, 2002.

[5] GARCES-ERICE, L., BIERSACK, E. W., ROSS, K. W., FELBER, P. A., URVOY-KELLER, G., Hierarchical Peer-to-peer Systems, in: Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par). Klagenfurt, Austria, 2003.

[6] KIRDA, E., GALL, H., FENKAM, P., REIF, G., MOTION: A Peer-To-Peer Platform for Mobile Teamwork Support, in: IEEE Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC'02), 2002.

[7] MELVILLE, L., WALKERDINE, J., SOMMERVILLE, I., (2003) D9 - P2P Reference Architectures (Final Version). Lancaster University, http://www.atc.gr/p2p_architect/brochure/0309F05_ReferenceArch.pdf

[8] MÝZRAK, A. T., Cheng, Y., KUMAR, V., SAVAGE, S., Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup, in: The Third IEEE Workshop on Internet Applications. San Jose, California, 2003.

[9] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., SCHENKER, S., A Scalable Content-Addressable Network, in: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM Press. pp. 161-172, 2001.

[10] ROWSTRON, A., DRUSCHEL, P., Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, in: IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pp. 329-350, 2001

[11] SCHOLLMEIER, R.: A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. In: 1st International Conference on Peer-to-Peer Computing (P2P 2001), Linköping, Sweden, IEEE Computer Society, 2001.

[12] YANG, B., GARCIA-MOLINA, H., Designing a Super-Peer Network, in: IEEE International Conference on Data Engineering. San Jose, California, 2003.