UNIVERSITATEA
BABEȘ-BOLYAI
Cluj-Napoca

FACULTATEA DE ȘTIINȚE
ECONOMICE ȘI GESTIUNEA
AFACERILOR

România
Ministerul Educației, Cercetării, Tineretului și Sportului
Universitatea Babeș-Bolyai
Cluj-Napoca
Facultatea de Științe Economice
și Gestiunea Afacerilor
Str. Teodor Mihali nr. 58-60
400591, Cluj-Napoca
Tel: 0264 418655
Fax: 0264 412570
E-mail: econ@econ.ubbcluj.ro

Babeş-Bolyai University
Faculty of Economics and Business Administration

Contributions to Collaborative Service Systems Implementation

Summary

Scientific Advisor:
Prof. Nicolae GHIŞOIU

PhD Candidate:
Ioan PETRI

Cluj-Napoca
2011

Contents

1

Introduction

The evolving complexity of collaboration introduced various methods of representations for data and information. Data and information identify a set of elements that can be put together as deliverable services to perform a useful activity: scientific instruments, processors, storage, network bandwidth, software and data. Current researching developments try to enrich collaboration with mechanisms for supporting service consumption, discovery and use. However, there is currently no accepted mechanism of collaboration whereby charges can be made for the use of these services and payments received. Such collaborative mechanisms are essential if current developed architectures evolve from an experimental status into professional or commercial use. By introducing effective lightweight mechanisms for accounting, charging and payment in the context of collaboration, an open market of services would emerge, co-ordinating users and providers of services. Whereas such a market would be global and electronically mediated, it is likely that many of the theoretical benefits of the free market would actually be realised, greatly encouraging the development and use of services [Hausheer and Stiller, 2007].

On the other hand, recent computational perspective has heightened the need for a powerful mechanism to control multiple interactions between entities. Non-traditional methods of collaboration have been identified in several researching attempts [Saito and Morino, 2010], [Cox and Noble, 2003], [Yang and Garcia-Molina, 2003] trying to offer reliable solutions for developing a local collaborative community. Within these solutions, responsibilities, roles, tasks, goals and communication are used to ensure a continuous process of collaboration. The new computation paradigm identifies a complex collaborative process between users in terms of sending and receiving large amounts of digital content and data. Although the amount of data can be limited, the increasing availability of resources and high bandwidth of Internet allow users to collaborate at different levels by communicating and sharing data and information. Service deliveries over the Web necessitate a well defined protocol and accurate communication. When working on Web, several important aspects must be accomplished such as trustworthiness, reliability, consistency, and availability of data. In addition, solutions for guaranteeing a level of service, expressing relevant QoS(Quality of Service) terms and supporting service collaboration are also needed.

Service Level Agreements have been validated as useful instruments for regulating the exchange of resources. Service level agreements (SLAs) are designed to determine the price for a service at an agreed quality level as well as the penalties in case of SLA violation. An SLA contains guarantee terms that need to be satisfied by a provider, and a payment that needs to be made by a user when such guarantees have been met. As an SLA represents provisioning that must take place in the future, an SLA may be circulated within a collaborative community. In that respect, a mechanism for regulating online markets by using the advantages of Service Level Agreements can bring a significant benefit for collaboration.

1.1 Research Questions

Considering a continuous process of adapting collaboration to the requirements of the current distributed computational paradigm we consider the following aspects:
Question 1: How a number of resources organised as deliverable services can be exchanged among users in Peer-to-Peer markets?
Analysis: The increasing availability of Internet services has stimulated the development of peer-to-peer markets. These electronic markets have the potential for improving the efficiency of trading by reducing search and transaction costs. An SLA can itself be used as a complementary currency to encourage service

exchanges between peers. Our work demonstrates how an SLA can be used as a medium of exchange and used to establish a market for computational resources. The value of an SLA can vary based on demand for particular types of resources.

Question 2: How market mechanisms can mediate the exchange of data and computational services between clients and providers?

Analysis: Within a market mechanism, complementary currencies can promote heavy competition between participants, and the removal of wealth from long term wealth holding structures (natural/material wealth, property, etc) to aid competitive innovation and better use of resources across various members of a community.

Question 3: What counter measures can be taken in order to prevent local communities from unpredictable behaviours?

Analysis: When a client selects provider(s) for establishing Service Level Agreements (SLAs), especially if clients have not directly interacted with particular providers, it is useful to identify the level of trust that can be placed in the selected provider. Trust can represent a reliable instrument to evaluate users based on previous activity filtering the community and excluding adversary users.

Question 4: How a local collaborative community can be modelled in accordance with the principles of market oriented systems for supporting exchanges between clients and providers?

Analysis: A collaborative community can group the overlapping and fragmented services produced and maintained by users from various ad-hoc communities, organisations, government authorities and official institutions. In general, these services are represented by text documents, spreadsheets, drawings, images, videos and databases such as product libraries. Such services are often maintained in systems with varying diversity and heterogeneity, making interoperability of systems a difficult challenge to realise in practice. Enabling a local market where various actors can participate and benefit can represent an added value for a community.

1.2 Contributions of this Research

An SLA may be used to specify quality of service terms, the measurement criteria, reporting criteria and penalty/reward clauses between participants. Within an electronic market, an SLA may be used for: (i) an expression/proof of debts as well as credits – debts to the client and credits to the service provider; (ii) as a token of exchange between participants(such as [Liebau et al., 2005]); (iii) as an identification of responsibilities of participants involved (such as the client and service provider). Facilitating the exchange of services has become an alternative solution for Peer-2-Peer (P2P) networks as powerful instruments to promote exchange. iWAT [Saito, 2006], Geek Credit [Komarov, 2004] and Ripple [Fugger, 2004] are examples of systems that use such complementary currencies for exchanges of computational service such as storage, computational capacity and network bandwidth. In a local system, participants can operate exchanges creating virtual markets. Within such a market, a peer node can exchange capability that it does not immediately need for an alternative service – thereby enabling a better sharing of overall capacity between participants in the market. In this thesis we identify how an SLA can itself be used as a complementary currency to encourage resource sharing between peers. We demonstrate how an SLA can be used as a medium of exchange and used to establish a market for computational and data services.

Virtual markets can present limitations in untrusted environments where various unexpected behaviours can happen increasing the risk associated with market exchanges. These aspects are addressed by proposing a trust model based on the aggregation of feed-backs. In more detail, our trust model tries to identify a set of malicious behaviours which are often met in online communities in order to ensure trustworthiness over the collaborative system. In our framework we consider both a client and a provider view – where compliance with an SLA for a provider can be measured, whereas clients provide feedback on their previous interactions to other clients (as a means of providing a recommendation). In this particular work we consider clients to have different types of behaviours (both truth telling and deception), whereby feedback about a particular provider may be influenced by particular incentives that a client may have. A key contribution of this work is to identify how malicious intent (based on incorrect feedback) can bias the overall trust establishment within a peer community of clients and service providers, and how trust values change with the number of clients involved in the community and with those providing feedback.

For applying virtual SLAs in a real local community, we develop a collaborative community on the basis of a service oriented architecture. We identify and demonstrate how discrete services can be deployed for further consumption and how collaboration can be ensured between various engineering organisations. Within the system each organisation can play the role of provider or client where clients and providers are represented by industry organisations from the engineering sector. The interaction between participants can identify two distinct operations:(i) provision – performed by providers when delivering various services and (ii) consumption – performed by clients when acquiring services.

1.3 Overview of the Thesis

The rest of the thesis is organised as follows:

In chapter 2, we analyse the process of collaboration identifying characteristics and related functional forms of collaborating. We provide an overview on how collaboration can be adapted to a distributed architecture and which are the characteristics of collaborative service systems. In particular, we tackle on how Peer-to-Peer systems can support collaboration and advantages of such approach are identified.

Chapter 3 tackles on the core principles of service oriented architectures. In particular, we focus on Service Level Agreements, performing an investigation of how the concept has evolved over time and which are the specification standards that can be used for modelling different service operations.

Chapter 4 describes the significations of complementary currency systems. We also perform an investigation on market oriented systems in order to identify which are the processes that complementary currencies can intermediate. We show how complementarity is a fundamental principle of market oriented systems and how complementary currency systems are applicable when dealing with local instances of oriented markets.

The general methodology is presented in chapter 5. In this chapter we present the simulation as a method for validating researching hypothesis. In particular, we explain the premises of adopting simulation for validating our hypothesis and which are the advantages of this approach when conducting a research process. An evaluation of peer-to-peer simulators and the characteristics of the PeerSim simulator are also part of this chapter.

The protocol for supporting the exchange of services in Peer-to-Peer markets and the findings of this approach are presented in chapter 6. This chapter represents the core contribution of this thesis where we demonstrate how an SLA can be used as a complementary currency for regulating P2P markets. Analysing related systems we present the advantages of SLAs for supporting exchanges in a market of computational and data services. Within this chapter, we explain how a local community where participants can issue their own payment objects can be constructed. Various scenarios are tested via simulation and related results are analysed in order to emphasise the benefits of the protocol.

A trust model for ensuring secure exchanges in Peer-to-Peer markets(discussed in chapter 6) is presented in chapter 7. In this chapter we investigate which are the advantages of trust for protecting a system against various type of adversary peers. We propose a trust model based on feedback aggregation to address the limitation of malicious behaviours that can occur in untrusted environments. Details about the methodology and findings of this research are also comprised in this chapter. In addition, we demonstrate how the proposed trust model is applicable for P2P clouds by applying the feedback aggregation mechanism for trust in solving the formation and usage of P2P ad-hoc clouds.

The applicability of the exchange protocol(6) in the context of a local engineering community is covered in chapter 8. In this chapter we develop a collaborative service system addressed to the engineering industry identifying the arguments for constructing an engineering market together with a set of objectives that must be accomplished by a collaborative service system. We present how a service oriented approach can be used for modelling a collaborative community with various participative actors from industry and education operating service exchanges.

Conclusions and future researching directions are presented in chapter 9.

2

Collaboration and Collaborative Systems

This chapter documents some collaborative approaches as a introductory hypothesis of the thesis. This chapter performs a survey on the most important aspects of collaboration trying to debate on how collaboration can be applied within a distributed environment. We identify operations to be performed in the context of collaboration together with a set of measures that need to be applied in order to adapt collaboration to various architectural contexts.

2.1 Collaboration Analogies

In the context of collaboration, technologies have evolved over the years and users have been obliged to change practices in order to access different applications and communication systems [Wold, 2005]. Collaboration [Javanmardi and Lopes, 2007] defines a process where different elements are combined with the ultimate objective of solving complex problems. According to the context that collaboration is meant to model and how the parties are scheduled to interact, collaboration can be deployed on different levels. The starting level is represented by a simple interaction between entities. It is important to note that in the context of the collaboration there are interactions which can happen independently. In order to identify the difference between interactions and the levels of collaboration, Grosz(1999) in AAAI Press tackles on the following arguments [Grosz, 1994]:

- The entities involved in an interaction have predefined objectives and can avoid building a certain functionality
- The interacting process is not meant to involve and use communication channels
- The results of the interacting process cannot be predicted

2.2 Collaboration Structure

From the perspective of collaborative elements and the subsets of keys involved in collaboration, the following categories can be distinguished [David Coleman, 2008]: (i) synchronous collaboration – a computer-mediated interaction between two or more entities identifying a 5 seconds interval, (ii) synchronous collaboration – a particular case of collaboration where the interaction has no time limitation and (iii) semi-synchronous collaboration – specifying intervals of interaction limited to 5 seconds.

Collaboration as a process can use the following levels:

- Communication
- Cooperation
- Coordination

Each level necessitates the completion of the targeted parameters. Communication defines an unstructured exchange of information, assumes a data transfer among entities and works as a tool for mediating the informational transfer.

2.3 Peer-to-Peer Collaborative Systems

Conventional collaborative systems use a centralised architecture where client requests are submitted to a server in order to retrieve needed resources. Such a centralised approach can work when the number of users and resources is limited. On the other hand, the collaborative systems can intermediate distributed entities to work together and eventually perform various operations on a shared content. Collaborative scenarios have recently evolved due to the proliferation of mobile devices with high-bandwidth network connectivity. The interaction between users has reached a level at which flexible and simple setup of provisional teams is a necessity. Users involved in collaboration need instant and ubiquitous access to services regardless of their current location. In addition, current collaboration deals with an increase necessity for effective means to communicate and cooperate. In relation with the current computing expansion, it must be noted that collaborative communities are today increasingly dynamic and subject to frequent changes.

In collaboration users need access to both required resources and available services. Services in the context of collaboration can define data and information such as personal calendar, files stored at the corporate server, or common data, or more complex services such as conference call support, notification of document changes or collective editing. When collaboration is deployed mechanisms such as distributed storage, and secure and authenticated access to services are needed. A significant attention is currently paid on large-scale distributed storage (making the Internet your hard-disk) and how this can be integrated in a collaborative process. Collaboration can be identified along various scenarios in dependance with the technological strategies supporting the collaboration: (i) a standard centralised solution, (ii) a pure P2P approach, and (iii) a mixed model comprising peer-nodes that can rely on some pre-existing infrastructure.

2.4 Summary

In this chapter, research is based upon various methods of collaboration and the associated mechanisms for adapting collaboration to different architectures. It was identified that collaboration becomes limited when applied in distributed environments. Therefore, an extension is required in order to perform collaboration in a dynamic distributed environment. Various solutions for peer-to-peer collaboration have been analysed and considered to be applied for supporting service deliveries. This chapter presents how a virtual collaboration engages entities with common interests to interact in various contexts achieving added value for community. In addition, when entities can contribute with services for achieving an objective, a strategic value and an economic impact of collaborative service system is implied. This impact can define an improvement in the level of community benefit and can develop the level of research within the explored context by creating new services. Thus, a collaborative service system stimulates the formation of long term collaborative relationship between entities.

3

Service Oriented Architectures and Service Level Agreements

This chapter presents the characteristics of service oriented architectures and associated advantages for modelling various contexts. The second part of this chapter is assigned to Service Level Agreements. We present a short history of the concept together with few important characteristics of electronic contracts. In addition, we identify the most used specification standards of Service Level Agreements presenting advantages and limitations for each standard.

3.1 Service Oriented Architectures

A service-oriented architecture(SOA) is essentially a collection of services. These services can communicate with each other and this particular communication can involve either simple data passing or more services coordinating a specific activity. SOA is achieved in dependence with Service Oriented Computing(SOC) and the associated middle-ware adaptation for service discovery, utilisation and combination of interoperable services for virtual and business process. SOC is a computing paradigm that organises and utilises services as central elements in order to enhance the development of distributed architectures in heterogeneous environments [Georgakopoulos and Papazoglou, 2009]. In the context of SOC, services can cooperate by creating dynamic business processes and applications addressed to various organisations. Service Oriented principles are technologically independent, the main function of applications being delivered as services. However, it was proved that a protocol for connecting services to each other is needed [Erl, 2005].

The term of *service* identifies a special resource representing a capability of performing tasks with a coherent functionality (provider entities and requester). Within the architecture, a service is considered to be a resource that can be transferred between two or more entities. At the same time, one service can perform one or more tasks and it can have a services description, a service interface, service semantics, and an identifier. A service has one or more roles and policies in relation to the service owner(provider). The policies being applied on each service are parts from the definition together with the entity owning a service and the entity receiving the service. Therefore, one service has an associated provider scheduled to deliver the service at a specific time and a requester(client) that can consume the service [Petri, 2009a]. Whereas the implementation of SOA is performed in relation with Web Services, the interface of a service can be described with Web Service Description Language (WDSL) or Extensible Markup Language (XML) [Paoli et al., 2008]. In addition, several other specification standards such as WS-BPEL, WS-Security, WS- ReliableMessaging can be considered applicable for supporting the service orchestration [Bianco et al., 2006].

3.2 Service Level Agreements

At the moment of the service selection, a third-party entity or an architect is expected to look for a specific set of services with a particular functionality. The decision is made according to a desired functional requirement or according to an qualitative attribute for evaluating the service. This quality is described within a service level agreement statement (SLA) as a contract which defines the level of service to be delivered. Service Level Agreements have been validated as efficacious instruments in establishing frameworks for the delivery of service among users or organisations. They first appeared in early 1990 as instruments for measuring and managing the quality of service (QoS) being delivered to clients from IT departments.

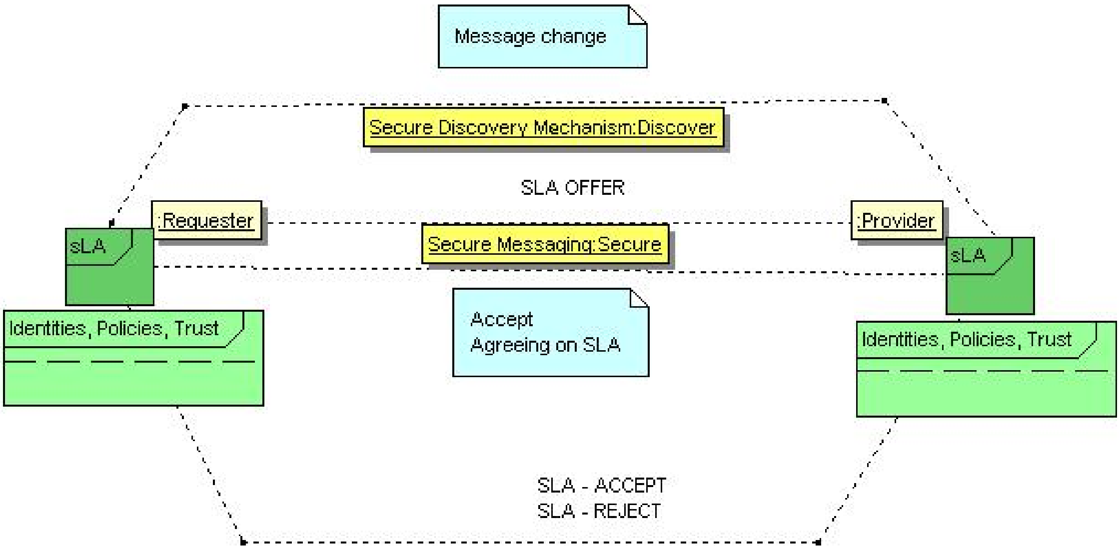

Figure 3.1: Service Level Agreement mechanism

A SLA function(see figure 3.1) contains a number of parameters designated to control the service operation. Current information systems proved as specialised in supporting continuous interaction of services and resources. They introduce several social and economic mechanisms such as virtual negotiations and service agreements in order to ensure the quality of service in the context of commercial infrastructures. Service markets are used as analogies with real markets where real products are replaced with various computing services. To simulate, real economies virtualisation is used as a current practice for exposing computational and data resources as services. The novel computing paradigm situates users as autonomous entities allowed to negotiate for specific functionality of the service they acquire. A service acquisition represents a common method for completing specific internal tasks for users. Many infrastructures are developed from the necessity of demand for computing resources because demand itself is a factor that constantly increases when new products and processes appear.

3.3 Summary

In this chapter a discussion of several principles of service oriented architectures has been presented. Here, the key aspects which are examined rely on the ability of service oriented architecture to provide means of access to distributed services, which are predominantly provided on a contractually basis. Service Level Agreements are analysed for their capacity of integrating with service oriented architecture and to bind users and resources that are spread across multiple administrative domains. It was identified among multiple specification languages that WS-Agreement standard is the most common method for specifying SLAs. Various contexts where SLAs can play an important role have been identified in order to introduce SLAs into a new virtual level of perception.

4

Market Oriented Systems in Practice

In this chapter the concept of market oriented computing is detailed along with a discussion of the intricacies needed to create a practical solution for supporting the exchange of services. The components and the mechanisms associated with market oriented systems are introduced together with significant economic processes involved in oriented computing. The second part of this chapter performs a survey of complementary currency systems and related processes engaged in local communities. Keeping a parallel to oriented computing we present how complementary currencies can be applied for developing a service oriented economy.

4.1 Market Oriented Computing

Once with the growing popularity of the Internet, electronic markets can bring together large numbers of buyers and sellers across different communities on significant scales changing the manner in which computing, communication, and business are managed. Powerful computers and high-speed networks are used as low-cost commodities which can be shared within a community. The new type of commodities have significantly changed the business paradigm; education, research, healthcare,economy, defense, and virtually all other human activities can benefit from these services and the information accessible through the Internet. The new computing and communication infrastructure deploys a service-based computing economy where various resources are mapped into deliverable services. Therefore, a service-based computing economy identifies a local commodity market for the provision and acquisition of computational and data resources.

To address these challenges, several researching attempts [Buyya et al., 2000], [Buyya et al., 2001], [Wolski et al., 2001], [Buyya et al., 2005], have proposed a distributed computational economy-based framework for regulating the supply and demand of the available services. This economy-based framework offers an incentive to service owners for rewarding the contribution and motivating users to enhance tradeoffs in relation with the processing time (e.g., deadline) and computational cost (e.g.,budget), depending on their QoS requirements.

4.2 Complementary Currency Systems

Complementary currencies generally describe a group of currencies or *scrips* designed to be used alongside standard currencies. They can be valued and exchanged in relationship to national currencies but also function as a medium of exchange on their own. Complementary currencies have also found applicability in Peer-2-Peer (P2P) networks as powerful instruments to promote exchanges – particularly to avoid the need for accessing central servers. Complementary currencies provide an useful abstract for supporting the use of computational and data services. The possibility of using complementary currencies for exchanging services within P2P networks have been investigated by many studies. Complementary currencies have different rates of exchange and scopes of circulation. The value of a complementary currency can change over time and in most instances is also related to the value of some real resource (such as commodities i.e. gold, oil, or services). Relationship of such a currency to a "service" is particularly interesting, as the value of the currency is based on the time required to perform a service in hours, notwithstanding the potential market value of the service. Such a currency is primarily used for mapping "human time" for performing a task into an economic value in a local context. Complementary currencies can also devalue over time (e.g. through the use of negative interest), to stimulate market exchange, thereby encouraging greater participation in the market. Devaluation also prevents storage of wealth (hoarding) of currency and encourages spending as the value continues to decay. Other experimental complementary currencies use high interest fees to promote heavy competition between participants, and the removal of wealth from long term wealth holding structures (natural/material wealth, property, etc) to aid competitive innovation and better use of resources across various members of a community.

4.3 Summary

In this chapter the notion of market is analysed from different perspectives. We tackle on the market process in association with market oriented paradigm and complementary currency systems. Therefore, the chapter starts with a survey on computing oriented economies insisting on relevant economic mechanisms such as pricing schemes and alternative principles of indivisibility and complementarity. Further, we identify how a market process can be achievable through the solution provided by complementary currencies, which can intermediate and regulate an alternative community. It important to note that notions such as markets, pricing, complementarity and currencies are useful and applicable for the work deployed in chapter 6 and 7 respectively.

5

Methodology

The approaches evaluated in chapter 2 and 3 aim to provide a better understanding on collaboration and local communities with associated market mechanisms. Finding the means to reproduce these processes can be a challenging tasks to overcome. This chapter presents the premises for deploying a successful researching method with respect to various operating environments to be modelled. Simulation is explored together with the facilities it provides for performing an authentic process of research. We explain why peer-to-peer simulators are useful when modelling a distributed environment especially when the environment is dynamic and identifies a high number of events to reproduce. The characteristics and the advantages of PeerSim simulator are provided in the second part of this chapter.

5.1 Simulation in Peer-to-Peer Systems

According to Banks et al. [Banks et al., 2000] simulation represents "the imitation of the operation of a realworld process or system over time". Simulation involves the development of a simulation model representing the system which is used to study the behaviour of this system over time, and to draw inferences concerning the operating characteristics of the real system. In a peer-to-peer context, the system is the real-world peer-to-peer network, and the model is the simulated peer-to-peer network. Banks et al. define several classifications of simulation models. A simulation model may be classified as static or dynamic, deterministic or stochastic, or discrete or continuous[Cassandras, 1993].

In large terms, P2P systems identify a distributed architecture where users have full autonomy. In particular, pure P2P architectures rely on decentralised services eliminating any centralised control mechanisms. In pure P2P architectures, users establish the amount of computing services to contribute in association with a predefined time interval. P2P architectures present high flexibility for handling large sets of users joining the network and offering robust mechanisms when users are leaving the network. P2P offers also a balancing mechanism of load across nodes, high flexibility, self-determination and low costs enhancing the number and the value of services provided by the system. A fundamental contribution of peer-to-peer systems research is to provide a reliable architecture for allowing users to collaborate over the Internet and efficiently accessing and exchanging services [Castro et al., 2002].

5.2 Portal-Based P2P System

Portal-based P2P systems allow the distribution of services among many peers on the basis of a decentralised architecture. Each peer in the network can provide services and can interact with other peers from the system. Portal based P2P systems offer users the facility to manage services within a P2P network. A Portal-based P2P system is designed to manage the interaction between peers. From figure 5.1, we observe the layer distribution of the portal. Portal peers can provide services within the portal with an associated interface, as well as they can provide services independently from the platform. It is important to note that the portal itself can be a service which can be accessed by external peers.

In order to provide greater applicability of the portal based P2P systems a simulator can be used to model various scenarios and validate hypotheses. Using the portal based abstraction a P2P simulator can be used to experiment different set-ups and configurations.

In chapter 8, we will exemplify the major contributions of this thesis be designing and implementing a portal-based P2P system for the engineering community. Thus, we will demonstrate the practical applicability and the real business need

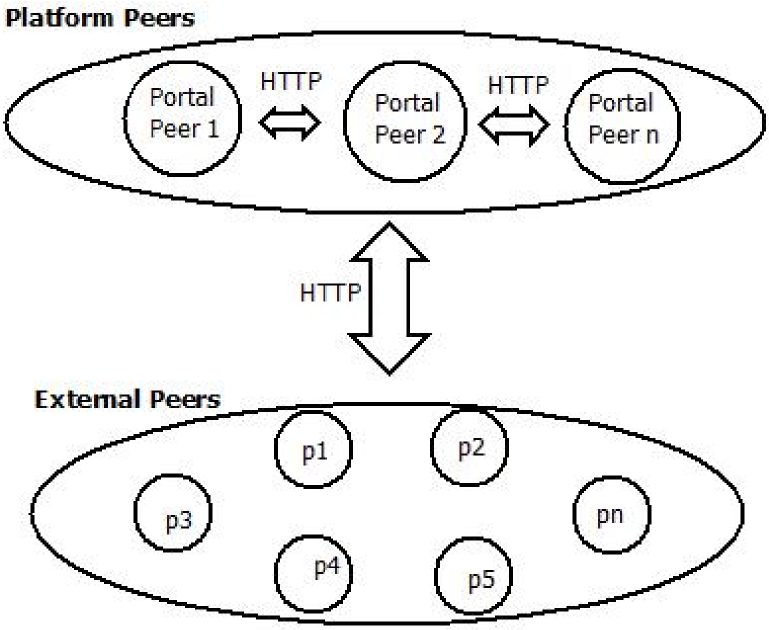

Figure 5.1: Portal-based P2P System

for our models. In chapter 8, we will exemplify the major contributions of this thesis be designing and implementing a portal-based P2P system for the engineering community. Thus, we will demonstrate the practical applicability and the real business need for our models.

5.3 PeerSim Simulator

Peer-to-peer (P2P) systems present two important features: (i) scalability and (ii) dynamism. Peer-to-Peer systems have appeared from necessity to reproduce realistic environments and simulate various researching scenarios. PeerSim identifies an extremely scalable simulation environment that supports dynamic scenarios. In PeerSim, the protocols need to be specifically implemented in relation with a predefined PeerSim Java API, or they can evolve into a real implementation [Jelasity et al., 2005]. PeerSim offers important modularity facilities as well as it is extremely flexible in terms of configuration. The network is modelled as a list of nodes where a node has a list of protocols, and the simulation has initializers and controls. PeerSim offers predefined components which can be used for simple experimental setups. For complex scenarios these components can be replaced with alternative implementations based on various preferences. The setup of the simulation process is represented as a plain text configuration file similar to a Java property file. These configuration properties can define implementations (Java classes) of components, and they also specify numeric or string parameters for these components. Configuration files define the properties of an experiment specifying needed parameters and associated values [Jelasity et al., 2007], [Biazzini et al., 2009].

5.4 Summary

An overview of simulation along with the main stages required for completing a reliable simulation process is provided within this chapter. There are two distinct parts which must be considered when discussing about simulation: (i) one identifies the process itself with all the required stages while (ii) the second one represents the environment to be modelled in the process. From the perspective of the environment to be reproduced, this chapter describes the peer-to-peer architecture alongside various generations of software. From the simulation point of view, simulators are evaluated identifying relevant aspects to be accomplished when deploying a simulation. In particular, the strengths of PeerSim simulator are presented together with compatible modules for reproducing the processes described in previous chapters.

6

A Model for Supporting Service Exchanges in Local Markets

The main contribution of this thesis is covered in this chapter. We perform a survey on the existent alternative systems tested in real live as well we analyse related complementary currency systems. As an SLA represents provisioning that must take place in the future, an SLA may be circulated within a local system generating revenues and regulating markets. The assumptions and the details of the protocol for supporting the exchange of data and computational services are provided within this chapter. We also deploy various experimental setups for testing the protocol and identify the overall benefit of the community.

6.1 Motivation

The notion of commodity money [Gessel, 1913] and the associated concept of "complementary currencies" have proved to be useful instruments to facilitate economic regeneration. Complementary currencies are recognised by their ability to add value to a community, and to improve the scalability of trade beyond bartering. In the context of a local economy (i.e. where trading is restricted to participants who are *nearby* – and does not involve global entities such as central banks), goods or services can themselves represent tradable objects. Complementary currencies generally describe a group of currencies or *scrips* designed to be used alongside standard currencies. They can be valued and exchanged in relationship to national currencies but also function as a medium of exchange on their own. Complementary currencies have also found applicability in Peer-2-Peer (P2P) networks as powerful instruments to promote exchange – particularly to avoid the need for accessing central servers. In P2P systems using SLAs we can use economic benefit as a metric to identify whether profit or loss has been incurred based on an initial configuration at start time. Several system configurations are experimented with in order to observe the reaction of the system in the context of different market scenarios. Our market simulation investigates how the issuing intervals can affect the overall benefit and how different SLAs types can induce profit or loss. Large scale systems are also evaluated for identifying the scalability of our approach. Within a P2P market a variation in demand for particular types of services can impact the *value* of the traded object. If the traded object is an SLA, the value of an SLA can decay or increase over time based on this demand fluctuation; consequently, we introduce a "welfare" parameter in order to measure the status of the system at each stage of exchange. The level of welfare can be aggregated across the entire P2P system, or a subset of nodes, and reflects the value benefit achieved by the system (or subset). We are also interested in identifying how differing levels of demand and system configurations cause a change to the level of welfare. In this particular contribution we focus on identifying the value of an SLA as the number of participants within a community increases, and utilize a PeerSim based simulation to show how community welfare can change over time.

6.2 Approach

An SLA may be used to specify quality of service terms, the measurement criteria, reporting criteria and penalty/reward clauses between participants. Within an electronic market, an SLA may be used for: (i) an expression/proof of debts as well as credits – debts to the client and credits to the service provider; (ii) as a token of exchange between participants(such as [Liebau et al., 2005]); (iii) as an identification of responsibilities of participants involved (such as the client and service provider). Establishing an SLA between two parties (client & service provider) implies that the service provider has agreed to provide a particular capability to the client within some quality of service. In return, the client must provide a monetary payment (most often) or credit to the provider once the service has been delivered (subject to a penalty, often also monetary, in case the quality of service terms have not been adhered to). The credit may be made at the beginning of service provision (the *pay-before-use* model) or after (the *pay-after-use* model). It is useful to note that an SLA refers to a service provision that must take place some time in the future. The SLA creation process starts when a client (the initiator) sends a request to a potential provider. The provider issues an SLA template, specifying agreement terms and obligations – containing service level objectives, quality terms and business values associated with particular service level objectives. Penalties and rewards are also parts of the SLA template. The initiator fills the template with the required service, asking the provider for a price. The agreement is finalised when the initiator accepts the price from the provider. The underlying protocol can be found in the WS-Agreement specification [Pichot et al., 2008]. In the context of an electronic market, the SLA expresses a commitment for a service delivery that is scheduled to start sometime in the future.

6.3 Summary and Conclusions

Exchanging services tends to be the next paradigm in terms of innovative informatics research. As users are more an more necessitating external services in order to cover internal requirements, previous solutions can become deprecated or limited when dealing with decentralised and dynamic environments.

This research aims to provide a better understanding on how an SLAs can be used as a complementary currency (exchangeable token) between participating nodes within a P2P network. As an SLA represents provisioning that must take place in the future, an SLA may be circulated within a community of trusted peers. This may occur either because a peer speculates that the capability offered by another peer may be at a higher demand at some time point in the future, or if a peer no longer needs the capability that it requires from another peer (but for which an SLA has already been generated). Trusted, in this context, implies that any client within the domain can redeem the SLA, and not necessarily the initiating peer. SLAs may also be used in this way by brokers for aggregating capacity from various peers within a network. In a pay-before-use model, the client has already paid the provider for the capability it has identified in the SLA. Therefore, rather than make an economic loss by abandoning the SLA, a client may decide to trade this on an open market to recover some of it's loss, or make a profit (if the demand for the service identified in the SLA increases after issuing).

Using simulation, we demonstrate that system benefit is influenced by the number of participants, the level of demand and the type of SLAs used. The experiments show that the view of peers (i.e. how many partners a peer can trade with), as a configuration parameter, induces benefit. Further, it was demonstrated that the system experiences a level of benefit directly related to the SLA submission interval – indicating that a higher circulation (i.e. lower submission interval) in the system improves overall benefit.

On the other hand, starting from the assumption that demand plays a major role in the welfare determination, we analyse the problem of fluctuation in SLA values over time. Therefore, we present the welfare level in the context of different demand configuration in order to expose the dependency between the demand of services and the system status in terms of welfare. We also discover that the heterogeneity of types (products) can represent an important factor that can induce new levels of welfare. It was also confirmed that the number of exchanged SLAs can be the cause for a variation in system welfare variation as the overall fluctuation of values is closely related to the level of welfare.

7

A Model of Trust for Service Exchanges

This chapter presents a trust model for addressing the limitation of exchanges(explained in chapter 6) performed in untrusted environments. With this model we target to identify how malicious intent (based on incorrect feedback) can bias the overall trust establishment within a peer community of clients and service providers, and how trust values change with the number of clients involved in the community and with those providing feedback. We simulate various scenarios and observe how the system reacts when dealing with malicious intentions.

7.1 Introduction and Approach

Electronic markets can bring together large numbers of buyers and sellers across different communities on significant scales. Hence, electronic markets have the potential for improving the efficiency of trading by reducing search and transaction costs. Online markets also allow buyers to choose the best possible deal for every transaction and interact with different sellers over time. This means that it is necessary to establish trust between buyers and sellers, especially where they may not have directly interacted with each other in the past. For online trading communities, the establishment of trust becomes a requirement for ensuring a secure environment for market exchanges.

Various views about trust exist within electronic communities and transactions. From one perspective, trust may be defined as a *subjective assessment* of influence within one system, changing perceptions about the quality and significance of a service [Kim, 2009]. Trust has an underlying *subjective* nature and may be used by one entity to control and sometimes manipulate other entities or groups. Trust may also be related to previous experiences which one entity has with another (based on previous interactions between the entities). Trust can also involve reciprocity, where one party can be morally obligated to give something in return for something received. However trust implies the necessity of a continuous evaluation in order to identify whether the interacting entity is dealing fairly or the level of reciprocity is adequate.

Trust can have different forms of representation in accordance with the mechanisms involved. For determining trust a variety of clues and past experiences are used to decide when such risk is appropriate. In addition, trust can be obtained by using indirect mechanisms such as social behaviours or third party experiences. Aggregating feedback and opinion about one entity (from a number of other entities) provides the reputation of an entity (considered as a community view about an entity) [Lik-Mui, 2002]. Reputation can also have an associated sanctioning role in social groups. When entities violate predefined trust standards they become subject to penalties. In the context of bilateral interactions involving risk, no stage can be performed until one party acquires a certain level of trust that can enable the second party to fulfill its obligations.

However, the study of trust outside formal mechanisms becomes more important in new communities where such mechanisms have yet to be firmly established. This is particularly the case for virtual (or electronic) communities today. These communities have created reputation or rating systems for the express purpose of encouraging trusting and trustworthy behaviours. For electronic markets, Service Level Agreements (SLAs) are efficient instruments for mediating business transactions between interacting entities – especially if such entities have not interacted in the past. An SLA may be used to specify quality of service terms, the measurement criteria, reporting criteria and penalty/reward clauses between participants involved in a transaction. Within an electronic market, an SLA may be used for: (i) an expression/proof of debts as well as credits – debts to the client and credits to the service provider; (ii) as a token of exchange between participants; (iii) as an identification of responsibilities of participants involved (such as the client and service provider). Establishing an SLA between two parties (client & service provider) implies that the service provider has agreed to provide a particular capability to the client within some quality of service. In return, the client must provide a monetary payment (most often) or credit to the provider once the service has been delivered (subject to a penalty, often also monetary, in case the quality of service terms have not been adhered to) [Petri et al., 2010]. When one party is unknown (provider or client) the level of risk associated with the transaction is considerably increased. Different studies [Arenas et al., 2010], [Lik-Mui, 2002], [Dingledine et al., 2003] have been investigating how a trusted environment can be developed by using reputation as a metric for monitoring the system.

In our framework we consider both a client and a provider view – where compliance with an SLA for a provider can be measured, whereas clients provide feedback on their previous interactions to other clients (as a means of providing a recommendation). In this thesis we consider clients to have different types of behaviours (both truth telling and deception), whereby feedback about a particular provider may be influenced by particular incentives that a client may have. A key contribution of our work is to identify how malicious intent (based on incorrect feedback) can bias the overall trust establishment within a peer community of clients and service providers, and how trust values change with the number of clients involved in the community and with those providing feedback.

7.2 Trust in P2P Systems

Within a P2P system there are several types of adversaries such as: traitors, colluding peers, front peers, white-washers, denial of service. In this study we offer a solution to defend against two important types of adversaries : (i) selfish peers and (ii) malicious peers. There categories are distinguished primarily by their goals within the system. Selfish peers try to use system services while contributing with minimal resources. A well-known example of selfish peers are "freeriders" [Hughes et al., 2005] identified within systems such as Kazaa and Gnutella.

The activity of malicious peers can affect members from the network or the system itself. For altering peer statuses, malicious peers can spend any amount of resources. Those malicious peers with constrained resources can be considered a subclass of malicious peers. As solution for malicious behaviours, reputation systems are usually designed to identify different types of adversaries. In addition, incentive schemes that enhance cooperation can be efficacious against selfish behaviours but ineffective against malicious peers. It has been identified that the number or the fraction of peers that are adversaries can impact the overall trust level of the system.

7.3 Summary and Conclusions

This chapter aims to provide a reliable mechanism for the trustworthy selection of partners for establishing Service Level Agreements. To address the limitation of untrusted environments we propose a model of selecting partners based on their trust level. We use an algorithm that enables peers to provide feedback about their direct (with a distance of one hop) or indirect (with a distance of multiple hops) neighbours. The algorithm uses as input the associated penalties and rewards from previous SLA exchanges and provides how the trust level changes as new participants enter the market (offering different types of services).

We also demonstrate how changes in behaviour (mailicious and truth telling) impacts the overall trust within the system. Malicious behaviour is controlled through two probability values that causes a peer to either provide incorrect feedback or alter its connectivity (referred to as migration). Simulating different scenarios by varying the probability values, we show that the level of trust within a system is closely related to the types of malicious behaviours existing within the service provider community. It was demonstrated that a complex malicious behaviour can significantly alter the level of trust whereas a simple malicious behaviour can only alter the trust level of a specific community. When simulating different malicious behaviours for dynamic systems, where the size of the network can vary, it is observed that large scale networks are more affected by malicious behaviours than small scale networks.

In addition we demonstrate how trust metrics may be used to support the reliable formation and usage of P2P based Cloud systems. P2P Clouds provide an alternative to data centre-based (often geographically centralised) Cloud systems available today. P2P Clouds enable end users to offer parts of their resources for use by others and in this way share similarities with volunteer computing systems – some of the differences are highlighted in [Neumann et al., 2011]. However, determining who can be trusted to provide resources that are reliable remains a significant challenge within such systems. Utilizing previous provision history (based on the adherence to or violation of an SLA) provides a useful basis for gauging reliability and therefore trust within a provider.

Hence, the selection of end users to contribute resources to a Cloud is based on their level of trust. Determining whether a provider is likely to be trustworthy is based on feedback from one hop neighbours; we demonstrate how such feedback can be both truthful and malicious (and where the behaviour of a peer may change from being truthful to malicious using a probability distribution), and the overall impact this distinction has on a Cloud system. The algorithm uses as input the associated penalties and rewards from previous SLA exchanges and identifies how the trust level is distributed within a P2P based Cloud community.

8

P2P-Based Collaborative Platform for the Local Engineering Community

This chapter aims to test in a real scenario the applicability of the protocols identified in chapter 6 and 7. This chapter demonstrates how the approaches presented in chapters 6 and 7 can be employed in a real scenario. Here, the scenario identifies a specific engineering community with various cases of collaboration and multiple institutional participants.

8.1 Motivation

The creation of advanced computing technologies for solving the complex challenges that face society and industry, including sustainability, in an innovative manner has become a dominant trend in recent years in engineering. The engineering industry needs to manage data that is fragmented across the participants involved at different stages of a project. Such data is often maintained in systems with varying diversity and heterogeneity, making interoperability of systems a difficult challenge to realise in practice. Ensuring means of access to fragmented resources represents a challenging task especially when users come from various disciplines such designers, architects, engineers, (structural, mechanical, electrical), facilities managers and clients. Data involved in engineering projects has generally included technical drawings and output generated from engineering applications.

We propose a model that provides means of access to data generated from various engineering projects. Our focus is on a specific collaborative model where private collaboration would be used to execute licensed, proprietary software, and store and manage project-specific data and models. Public collaboration could be used to support multiple versions of a software program which can be deployed and distributed as separate applications [Kumar and Cheng, 2010]. The proposed collaborative model addresses the industry by using various techniques such as ontology engineering, knowledge mining, information retrieval and service oriented computing. The system plays a pivotal role enhancing collaboration between various industry and researching institutions from the engineering sector. This generic collaboration principle holds a set of well defined principles and strategies which have to be followed; while for researching institutions involved within the system, the main objective is to enable a large scale dissemination of lessons learnt from the research activity, for industrial institutions the objective is to obtain an added value (in terms of information or profit).

8.2 Modelling Exchanges within the Community

The engineering community identifies a set of participants such as industrial and educational institutions sharing common objectives and performing various operational tasks. Each institution works with a set of objectives and targets to achieve a certain profit. Furthermore, each institution owns large sets of data and information relevant for the engineering community. These data sets can be shared among participants as deliverable services, creating a local economy with an associated market. This market can be developed on the basis of complementary currency systems where alternative currencies are used to regulate the market. Supply and demand determine the value of each service and the benefits of each participant.

For ensuring reliable functionality, the community is organised on the principles of portal-based P2P systems(presented in 5.2). According to this designing template, each organisation is represented as a portal peer which can perform various operations with services. Portal peers can provide services within the portal, with an associated interface or they can provide services independently. It is important to note that the portal itself can be a service which can be accessed by external peers.

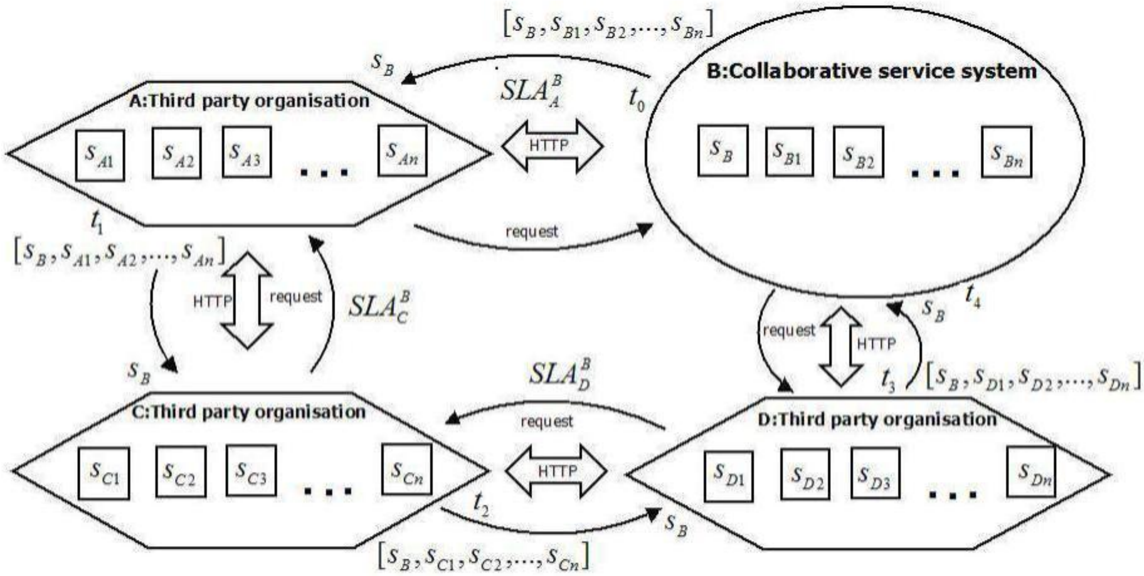

Figure 8.1: Modelling exchanges within the community

From figure 8.1 is observed a community with four different organisations. Organisation B represents the collaborative service system described in this chapter, while the organisations A, C, D define third party organisations from the engineering community. From the list of services that one entity can provide $S = [s_1, s_2, s_3, ..., s_n]$, a subset $S_D = [s_1, s_2, s_3, ..., s_m]$, $S_D \in S$ contains services that one entity can provide directly(from its own capability). The subset $S_I = [s_1, s_2, s_3, ..., s_p]$, $S_D \in S$, $S_I = S - S_D$, identifies a collection of services that one entity can provide indirectly(acquired from previous exchanges). Noting that each s_i is an SLA template defining terms and conditions of the agreement between parties, we illustrate how the exchange occurs over the community on the premises of complementary SLAs. Briefly, the process starts when institution A points a service request to institution B. In response, B replies with a list of deliverable services such as $[s_B, s_{B1}, s_{B2}, ..., s_{Bn}]$. Institution A chooses to acquire the service described in template s_B. This identifies the stage when the virtual SLA currency is issued and it circulates along the interval $[t_0 - t_4]$ over the community.

8.3 Summary and Conclusions

Local communities have been validated as efficacious for their capacity to produce revenues for participants. In such communities the collaboration is simplified and an added value is achieved periodically. We propose a model of collaborating with respect to service oriented architecture where various engineering resources are organised as discrete services which can be accessed within the community. The system models the context of engineering as a local community where actors can deploy and consume services. These services can be provided on the single provision capabilities basis(each entity can deliver a single service) as well as multiple provision mechanisms can be applied. Starting from requirements we identify a high necessity for a virtual context in engineering where users such as institutions and industry organisations can collaborate at different levels. For addressing high demand for online collaboration in the engineering we apply a protocol which supports the exchange of services in local communities and identify several benefits of this approach.

9

Conclusion and Future Work

In this final chapter, the focus is drop to the research questions and to the research findings of this study. Section 9.1 presents a summary of the results within the thesis alongside with related discussions. Following this discussion, Section 9.2 tackles on possible researching directions triggered by this research.

9.1 Conclusions

The research presented in this thesis has been centred around non-traditional methods of collaboration in distributed environments. In such context we identify how an SLA can be used as a complementary currency (exchangeable token) between participating nodes within a P2P network. As an SLA represents provisioning that must take place in the future, an SLA may be circulated within a community of trusted peers. This may occur either because a peer speculates that the capability offered by another peer may be at a higher demand at some time point in the future, or if a peer no longer needs the capability that it requires from another peer (but for which an SLA has already been generated). Trusted, in this context, implies that any client within the domain can redeem the SLA, and not necessarily the initiating peer. SLAs may also be used in this way by brokers for aggregating capacity from various peers within a network. In a pay-before-use model, the client has already paid the provider for the capability it has identified in the SLA. Therefore, rather than make an economic loss by abandoning the SLA, a client may decide to trade this on an open market to recover some of it's loss, or make a profit (if the demand for the service identified in the SLA increases after issuing). Using simulation, we demonstrate that system benefit is influenced by the number of participants, the level of demand and the type of SLAs used. The experiments show that the view of peers (i.e. how many partners a peer can trade with), as a configuration parameter, induces benefit. Further, it was demonstrated that the system experiences a level of benefit directly related to the SLA submission interval – indicating that a higher circulation (i.e. lower submission interval) in the system improves overall benefit.

We extend the approach, with the assumption that demand plays a major role in the welfare determination by analysing the problem of fluctuation in SLA values over time. Therefore, we present the welfare level in the context of different demand configuration in order to expose the dependency between the demand of services and the system status in terms of welfare. We also discover that the heterogeneity of types (products) can represent an important factor that can induce new levels of welfare. It was also confirmed that the number of exchanged SLAs can be the cause for a variation in system welfare variation as the overall fluctuation of values is closely related to the level of welfare.

The exchanges in untrusted environments are enabled by proposing a model of selecting partners based on their trust level. We identify how trust can be applied in the context of enclosing Service Level Agreements as well as we demonstrate the applicability for peer-to-peer democratic clouds. In the algorithm, peers are enabled to provide feedback about their direct (with a distance of one hop) or indirect (with a distance of multiple hops) neighbours. The algorithm uses as input the associated penalties and rewards from previous SLA exchanges and provides how the trust level changes as new participants enter the market (offering different types of services). Simulating different scenarios by varying the execution probability we show that the level of trust within a system is closely related with the types of malicious behaviours. We also validate that a complex malicious behaviour can significantly alter the level of trust whereas a simple malicious behaviour can only alter the trust level of a specific community. When simulating different malicious behaviours for dynamic systems, where the size of the network can vary, it is observed that large scale networks are more affected by malicious behaviours than small scale networks.

In the extension of this trust model, we demonstrate how trust metrics may be used to support the reliable formation and usage of P2P based Cloud systems. This is achieved by utilizing previous provision history (based on the adherence to or violation of an SLA) provides a useful basis for gauging reliability and therefore trust within a provider. Identifying when a provider is likely to be trustworthy is based on feedback from one hop neighbours; we demonstrate how such feedback can be both truthful and malicious (and where the behaviour of a peer may change from being truthful to malicious using a probability distribution), and the overall impact this distinction has on a Cloud system.

In order to provide an applicable context where a number of organisations can exchange services, we develop a local collaborative system addressed to the engineering community. The participants within the system come from various engineering disciplines such as designers, architects, engineers, facilities managers and clients. The content to operate comes from data involved in engineering projects that has generally included technical drawings and output generated from engineering applications. Organising this content on the basis of service oriented architecture and identifying a set of market requirements we propose an architectural model for supporting the exchange of services.

9.2 Future Work

Complementary currency systems have been analysed by related studies from the prospective of different strategies that a peer can apply during the exchange. Various strategies can be identified at the peer level – in the context of untrusted peer-to-peer exchanges:

- Maximising social welfare in reselling actions.
- Profit maximisation when peers behave selfishly and try to maximise their profit from reselling, arbitrage or hedging.
- Satisfying demand by buying the cheapest SLAs.
- Increasing the size of the trusted network by purchasing forwarded SLAs that originate from unknown providers.

As peers can have different unpredicted behaviours which can reduce the overall benefit of the system the following actions have been proposed: (i) Elimination –Always try to use an SLA that a (trusted) partner peer has issued if there is one; (ii) Stretch – Always try to receive an SLA whose chain of endorsement is longer than those of others; (iii) Matchmaking – Prefer selecting a partner among the issuers of acquired SLAs;(iv) Forwarding – Always try to use an SLA whose loss of value will be greater than those of others if not used at the present time; (v)Deferring – Always try to avoid using an SLA against its issuer if the variance of its value over time has not stopped [Saito and Morino, 2010]. Identifying specific behaviours in untrusted peer-to-peer exchanges and associated strategies addressing these behaviours can represent a future investigation.

On-line service delivery undertaken between clients and service providers often incurs risks for both the client and the provider, especially when such an exchange takes place in the context of an electronic service market. For the client, the risk involves determining whether the requested service will be delivered on time and based on the previously agreed Service Level Agreement (SLA). Often risk to the client can be mitigated through the use of a penalty clause in an SLA. For the provider, the risk revolves around ensuring that the client will pay the advertised price and more importantly whether the provider will be able to deliver the advertised service to not incur the penalty identified in the SLA. This becomes more significant when the service providers outsource the actual enactment/execution to a data centre – a trend that has become dominant in recent years, with the emergence of infrastructure providers such as Amazon.com. A possible solution for addressing these risks can rely on fault tolerance techniques. In fault tolerance, various services can be replicated to different machines reducing risks and ensuring proper service deliveries. In particular, the mechanism of quorums [Lamport et al., 1982], [Birman, 2005] can be applied identifying each machine status in order to perform a secure provision of resources [Petri, 2009b].

From the perspective of electronic contracts [Rana and Ziegler, 2010] several researching aspects can be notable. The first is to identify what should be part of an SLA and how should this be encoded in terms of rules, constraints, keywords. This aspect is very useful from the perspective of establishing the exact metrics that should be included within an SLA. On the other hand it can be useful to identify how can be an SLA used after completing the definition process. One SLA contains a price and also a service distribution that can be transferred among different users. This becomes a subject in the context of Cloud Computing when users can share resources [McKee et al., 2010]. As an SLA specifies a particular service, can be useful to clarify which are the types of SLA that encourage the resource sharing. It can happen that users present more interest for operating specific types of service and consequently specific types of SLA. From the business perspective it might be interesting to consider the possibility of mapping business policies into operational SLAs. It is important to note that SLAs are useful when working with trusted environments [Rana et al., 2008]. One SLA can put together parties in order to perform service operations but it can show drawbacks when the participants are un-trusted. Thus, a better understanding of how to integrate trust metrics within SLA could be a really useful approach.

References

Arenas, A. E., Aziz, B., and Silaghi, G. C. (2010). Reputation management in collaborative computing systems. *Security and Communication Networks*, 3(6):546–564.

Banks, J., Nelson, B., and Carson, J. (2000). *Discrete-Event System Simulation. 3rd Edition.* Prentice Hall.

Bianco, P., Lewis, G. A., and Merson, P. (2006). Service level agreements in service-oriented architecture environments. Technical report, Software Engineering Institute, Carnegie Mellon University.

Biazzini, M., Banhelyi, B., Montresor, A., and Jelasity, M. (2009). Distributed hyper-heuristics for real parameter optimization. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO.* Springer.

Birman, K. P. (2005). *Reliable Distributed Systems and Technologies.* Springer.

Buyya, R., Abramson, D., and Giddy, J. (2000). An economy driven resource management architecture for global computational power grids. In *Proceeding of the International Conference on Parallel and Distributed Processing Techniques and Appplications (PDPTA)*, pages 26–29.

Buyya, R., Abramson, D., and Giddyet, J. (2001). *A case for economy Grid architecture for service oriented Grid computing.* Grid and cooperative computing, San Francisco, CA, proc 10 th ieee international heterogeneous computing workshop, ipdps edition.

Buyya, R., Abramson, D., and Venugopal, S. (2005). The grid economy. *Proceedings of the IEEE*, 93(3):698–712.

Cassandras, C. (1993). Discrete event systems. Aksen Associates, Boston.

Castro, M., Druschel, P., Ganesh, A., Rowstron, A., and Wallach, D. S. (2002). Secure routing for structured peer-to-peer overlay networks. *SIGOPS Oper. Syst. Rev.*, 36:299–314.

Cox, L. P. and Noble, B. D. (2003). Samsara: honor among thieves in peer-to-peer storage. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, SOSP '03, pages 120–132, New York, NY, USA. ACM.

David Coleman, S. L. (2008). *Collaboration 2.0: Technology and Best Practices for Successful Collaboration in a Web 2.0 World*. Happy About, California, USA.

Dingledine, R., Mathewson, N., and Syverson, P. (2003). Reputation in p2p anonymity systems. In *In Workshop on Economics of Peer-to-Peer Systems*, Berkeley, California.

Erl, T. (2005). *Service Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall.

Fugger, R. (2004). Money as IOUs in Social Trust Networks and a proposal for a Decentralized Currency Network protocol. Available at `http://ripple.sourceforge.net/`.

Georgakopoulos, D. and Papazoglou, M. (2009). *Service-Oriented Computing (Cooperative Information Systems Series)*. MIT Press.

Gessel, S. (1913). *The Natural Economic Order*. The Free Economy Publishing Co. Also available at: `http://www.appropriate-economics.org/ebooks/neo/neo2.htm`, Last accessed: June 2010.

Grosz, B. J. (1994). *Collaborative Systems Advancement of Science and the American Association for Artificial Intelligence*. AAAI Press.

Hausheer, D. and Stiller, B. (2007). Peermart: Decentralized auctions for bandwidth trading on demand. *ERCIM News*, (68):42–43.

Hughes, D., Coulson, G., and Walkerdine, J. (2005). Free riding on gnutella revisited: The bell tolls? *IEEE Distributed Systems Online*, 6:1–.

Javanmardi, S. and Lopes, C. V. (2007). Collaborative computing. In *Networking, Applications and Worksharing*, pages 299–302.

Jelasity, M., Montresor, A., and Babaoglu, O. (2005). Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems*, 23(3):219–252.

Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.-M., and van Steen, M. (2007). Gossip-based peer sampling. *ACM Trans. Comput. Syst.*, 25.

Kim, W.-S. (2009). Effects of a trust mechanism on complex adaptive supply networks: An agent-based social simulation study. *Journal of Artificial Societies and Social Simulation*, 12(3):4.

Komarov, A. (2004). Geek credit. Available at `http://home.gna.org/geekcredit/`.

Kumar, B. and Cheng, J. C. P. (2010). Cloud computing and its implications for construction it. In *Computing in Civil and Building Engineering*, Nottingham, UK.

Lamport, L., Shostak, R., and Pease, M. (1982). The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4:382–401.

Liebau, N., Darlagiannis, V., Mauthe, A., and Steinmetz, R. (2005). Token-based accounting for p2p-systems. In Muller, P., Gotzhein, R., Schmitt, J. B., and Brauer, W., editors, *Kommunikation in Verteilten Systemen (KiVS)*, Informatik aktuell, pages 16–28. Springer-Verlag.

Lik-Mui (2002). *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks.* PhD thesis, Massachusetts Institute of Technology, Massachusetts. PhD thesis.

McKee, P., Steve, T., Mike, S., and Lowe, R. (2010). *Market-Oriented Grid and Utility Computing–SLAs, Negotiation, and Challanges*. Wiley.

Neumann, D., Bodenstein, C., Rana, O. F., and Krishnaswamy, R. (2011). *STACEE: Enhancing Storage Clouds using Edge Devices alongside ICAC*. ACM Press, Karlsruhe, Germany, Proceedings of the ACM/IEEE workshop on Autonomic Computing in Economics (ACE) edition.

Paoli, T., Sperberg-McQueen, J., E., C. M. M., and F., Y. (2008). *Extensible Markup Language (XML) 1.0*. World Wide Web Consortium.

Petri, I. (2009a). Policies and identity in future internet. *, Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques–Knowledge Processing in Economics, Studia Universitatis*, Knowledge Processing:63–66.

Petri, I. (2009b). Quorums systems as a method to enhance collaboration for achieving fault tolerance in distributed systems. *Economy Informatics Journal*, 12(3):68–75.

Petri, I., Rana, O., and Silaghi, G. C. (2010). SLA as a Complementary Currency in Peer-2-Peer Markets. In *in Proceedings of the 7th International Workshop on the Economics and Business of Grids, Clouds, Systems, and Services, GECON 2010*, volume 6296, pages 141–152. Springer.

Pichot, A., Woldrich, O., Ziegler, W., and Wieder, P. (2008). Dynamic sla negotiation based on ws-agreement. In *International Conference on Web Information Systems and Technologies (WEBIST)*. Springer, proceedings of webist (1) edition. pp 38-45.

Rana, O., Warnier, M., Quillinan, T. B., and Brazier, F. (2008). Monitoring and reputation mechanisms for service level agreements. In *Proceedings of the 5th international workshop on Grid Economics and Business Models*, GECON '08, pages 125–139, Berlin, Heidelberg. Springer-Verlag.

Rana, O. and Ziegler, W. (2010). Research challenges in managing and using service level agreements. In Desprez, F., Getov, V., Priol, T., and Yahyapour, R., editors, *Grids, P2P and Services Computing*, pages 187–200. Springer US. 10.1007/978-1-4419-6794-$7_1$5.

Saito, K. (2006). *i-WAT: The Internet WAT System – An Architecture for Maintaining Trust and Facilitating Peer-to-Peer barter Relations*. PhD thesis, School of Media and Governance, Keio University.

Saito, K. and Morino, E. (2010). The brighter side of risks in peer-to-peer barter relationships. *Future Gener. Comput. Syst.*, 26:1300–1316.

Wold, T. (2005). Collaborative solutions - True Collaboration as the Most Productive Form of Exchange. In *Collaborative Solutions Newsletter*. Tom Wolff and Associates, Collaborative Solutions Newsletter edition. available at :`http://www.tomwolf.com/collaborative-solutions-newsletter-summer-05.htm`.

Wolski, R., Plank, J. S., Brevik, J., and Bryan, T. (2001). Analyzing market-based resource allocation strategies for the computational grid. *Int. J. High Perform. Comput. Appl.*, 15:258–281.

Yang, B. and Garcia-Molina, H. (2003). Ppay: micropayments for peer-to-peer systems. In *Proceedings of the 10th ACM conference on Computer and communications security*, CCS '03, pages 300–310, New York, NY, USA. ACM.