A stochastic Reputation System Architecture to support the Partner Selection in Virtual Organisations

Zur Erlangung des akademischen Grades eines Doktors der Wirtschaftswissenschaft
(Dr. rer. pol.)
von der Fakultät für
Wirtschaftswissenschaften
der Universität Karlsruhe (TH)

genehmigte

DISSERTATION

von
Dipl.-Math.(techn.) Jochen Haller

Tag der mündlichen Prüfung: 30. April 2009

Referent: Prof. Dr. Christof Weinhardt
Koreferent: Prof. Dr. Rudi Studer

2009, Karlsruhe

ABSTRACT

Virtual Organisations (VOs) are temporary alliances of globally dispersed, otherwise sovereign, organisations that pool their resources to jointly tackle a business opportunity that can not be addressed by one organisation alone. eBusiness in such a complex, evolving environment as the one encountered in VOs can only prosper with an intrinsic trust model, supporting various classes of VOs from different application domains throughout their entire lifecycle. Application domains range from dynamic, short-lived aggregated service provisioning VOs to more stable, long-lived collaborative engineering VOs. Business collaborations in Virtual Organisations (VOs) demand sound security infrastructures to achieve acceptance from industry. It becomes apparent that in dynamic environments requiring swift decisions as well as quickly adapted and connected applications, traditional hard security e.g. based on static access control is no longer able to cope with security alone to meet the collaboration objective. Soft security based on trust management, e.g. provided by a reputation system, is able to address those requirements dealing with previously unknown entities and hereby complement, but not replace, established static hard security. This work addresses the benefit of trust based decision support within VOs. In particular, reputation systems support a VO manager in the early VO phases by providing automated decision support to select the most reliable, trustworthy partner. Of course, the support is not limited to early phases but trust based controls may be established throughout the VO's operational for its members. This contribution delivers a stochastic trust model as the basis for a Stochastic Reputation System, abbreviated STORE. Trust is defined as a trustee's probability to behave reliably within expectation of a trustor. Such behaviour is classified and measured through so-called Trust Indicators (TIs), inherently characterising an organisation's trusted behaviour in VO relationships. A stochastic trust model, based on probability distributions, also caters for predictions of an organisation's future trustworthiness and captures the uncertainty of trust in particular in predictions. A further contribution of this work consists of an attack classification on reputation systems focusing especially on STORE set in a VO environment. The STORE model and architecture are finally evaluated in agent based simulations. As a result, STORE is able to provide the desired decision support for VO member selection. The STORE model quickly reflects dynamic changes in a VO members trustworthiness and allows to distinguish between organisations that specialise in a particular VO application scenario. Finally, STORE proves to be resilient against most of the possible reputation system attacks by its design.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

With the advent of increasingly dynamic business relationships, the demand for automation and swift set up of supporting Information and Communication Technologies (ICT) becomes more prominent. Focusing on electronic business or short eBusiness on a global scale, the Virtual Organisation (VO) is an emerging organisational structure incorporating dynamic Business-to-Business (B2B) relationships. A VO is defined as a temporary alliance of otherwise independent, geographically dispersed organisations or individuals which pool their resources to achieve a business goal one alone can not master [27, 46, 80, 110]. VOs can be classified according to their properties such as, among others, their expected lifetime, turnover and the application domain they take place in. On the one end range VOs from aggregated services provisioning, e.g. Wireless LAN (WLAN) roaming from the telecommunications application domain that are dynamic and short-lived. Stable, long-lived VOs from manufacturing such as the aerospace segment range on the other end of this classification. A VO follows a phased lifecycle where, especially in the initial set up phases, speed is of essence. Taking the large financial investments for VOs especially long-lived ones situated in the aerospace industry into account, protection of assets and therefore security is an important requirement. Traditional security measures such as authentication and access control offer the possibility for an organisation to expose its assets in a controlled fashion to other VO members. Assets are resources such as Business Processes (BPs), services or data. Setting up such security measures, however, requires a considerable ICT integration across the collaborating VO member's administrative domains. Such an integration typically amounts for 30-40% of the overall IT costs [76]. To avoid spending this money in vain, the integration must only take place with trustworthy partners that are carefully selected in the VO's initial phases. In this context, trust management contributes decision support as a security measures in its own right, complementing traditional security. They are not intended to replace traditional or hard security measures. Therefore, Rasmusson et al [92] denoted trust related security measures as soft security measures that are implemented to reduce risks, for instance the risk to select an untrustworthy VO member who needs to be replaced during later operational VO phases. Introducing more details about VOs and their application domains requires a deeper understanding of certain essential properties, e.g. a VO's life cycle. These fundamentals are presented in Subsection 2.3. A more detailed VO scenario description and problem definition is therefore postponed to the following Subsection 2.3.1.

Trust management and trust itself became popular topics in various research domains ranging from psychology over economics to computer science, including Information (IS) and Computer Science (CS). Trust itself unites cross-domain research fields ranging from social and economic sciences to technical fields. Therefore, trust is a demanding research topic and inherently difficult to be formally defined for usage in IT supported environments [44]. However, such a formal definition is required to employ trust in the digital domain.

It is desirable to apply trust in VOs to:

- provide swift automated decision support for selecting the most trustworthy VO members for a business purpose.

- predict the trustworthiness of their future behaviour.
- provide trust decision support throughout a business relationship following a Business Process.

With these trust applications in mind, Gambetta's well established definition of trust [44] is adopted for this line of work. He defined trust as the subjective probability by which a trustor expects a trustee to behave reliably.

The definition of trust as a probability by itself motivates the approach of designing a stochastic trust model taken in this thesis.

1.1 Problem Definition

In VO scenarios, trust is an important requirement for the collaboration across multiple administrative domains [57], starting with the partner selection process. Looking at an example from the Telecommunication application, which will be analysed deeper in the following chapters, the importance of the requirement becomes apparent. In this example, a VO forms to address the needs of a business traveller, consisting of a large number of organisations. The traveller desires to access electronic services through a mobile device, delivering local information such as entertainment or weather forecast services, but also her regular services provided by a home network operator, offering stock and other business information. Such VOs consist on the one hand of a large number of partners, ranging from access point over network to electronic content service providers. Especially the latter become increasingly numerous with the advent of Service Oriented Architectures, allowing to combine their service offerings in service compositions or so-called mashups. On the other hand such a VO must form very rapidly, in seconds, since the traveller is not willing to wait for information. A coordinator or manager of a telecommunications VO is now faced with the problem which partner to select from a large set of service providers, some of them offering the same required services. In such a dynamic market, the fluctuation of newcomers entering the market and unsuccessful organisations leaving is ever increasing. The manager can no longer rely on past experiences with the providers, there are no past observations and similar historic data available anymore.

Summarising the problem, the manager of a VO faces a decision problem. How should he select the most trustworthy partners in such dynamic environments with frequently changing actors and without prior knowledge of past performances? Trust and reputation management are able to provide decision support for VO environments. The stated trends of increased formation speed and frequently changing actors will require increased automated reputation based decision support in partner selection and management in VOs. This thesis addresses the stated overarching research question following up on the following focused topics:

- Can a reputation system separate specialised organisations of similar trustworthiness by reputation to provide decision support for a particular application domain? Trust, as a subjective property, is perceived differently in varying application domains.
- Can a reputation system cater for the dynamic trust aspects such as organisations changing their trustworthy behaviour over time? Trust, as a dynamic property, changes over time.
- How can a robust reputation mechanism and system be provided? Reputation systems can be attacked and exploited in many ways. A reputation system must be resilient against those attacks in order to provide meaningful decision support.

With respect to the actors, this work focuses on organisations playing the roles of trustor and trustee. Trust should be managed by a dedicated subsystem to contribute to the on-demand creation of the VO, quickly reacting to an emerging business opportunity. Reputation systems are such subsystems that integrate trust management and offer reputation information to a trustor about a trustee. In contrast to the subjective property trust, reputation strives to be an objective value. To achieve this goal, a reputation value is frequently defined as the result of an aggregation of multiple subjective trust values.

Trust and its evaluation is already context sensitive, therefore is reputation. Castelfranchi et al. state in [38] that "Trust basically is a mental state, a complex attitude of an agent x towards another agent y about the behaviour/action relevant for the result (goal) g". For the purpose of trust and reputation management in VOs, the goal sets the context trust and reputation are evaluated in. VOs are goal driven organisational structures, their goal is to swiftly address an emerging business opportunity. In collaborative engineering for example, this relates to the goal of assembling a consortium of specialised organisations to address a published tender for a plane upgrade. One criterium for selecting these organisations is their trustworthiness with respect to the VO's goal. Trustworthy organisations from the ad-hoc service provisioning domain are not necessarily considered trustworthy within the collaborative engineering domain as well, e.g. if they lack internal stability and expand to regions with low quality infrastructure. If the member organisations of a VO environment are supposed to accept a reputation system's automated decision support, there has to be trust into this system itself. This entails trust into the deployed reputation system and its owner, e.g. if the system meets network and system security requirements or if the owner acts as a trusted (third) party, but also if the reputation mechanism can be trusted. Feedback based reputation systems with a mechanism aggregating received feedback information towards a reputation value can frequently be cheated in different ways by malicious parties. To achieve trust into a reputation mechanism, it has to be resilient against known general attacks on reputation systems as well as possibly emerging domain specific attacks.

Besides the stated non-functional requirements for the underlying trust model, a reputation system for VOs has to meet ICT imposed functional requirements as well. VOs, by definition, require the collaboration of their members and the time to become operational is paramount. A swift VO set up can only be achieved by relying on the member's integration of their ICT infrastructures, building on open, standardised interfaces and protocols. The Service Oriented Architecture (SOA) paradigm is now a well accepted concept in industry. Functionality and logic are encapsulated in small, modular and reusable services that are all building blocks of a SOA. The Web Service stack is the most followed set of technical SOA standards with industry backing. A reputation system as a subsystem in such an environment has to follow SOA architecture design principles by offering its functionality as a reputation service.

Addressing the stated problems and resulting requirements, the thesis proposes a novel trust model as a reputation system's core. It is designed for use in a VO environment, to provide automated decision support for selecting the subset of the most trustworthy VO members from a larger set of potential VO members and observe their trustworthiness during VO operations. The reputation system is designed to fit into an ICT environment following SOA principles.

The thesis's main contributions are the following:

1. A trust model catering for the following properties:
   (a) **Directed Relationship** - Trust is a bidirectional relationship between a trustor and one or more trustees.
   (b) **Subjective** - Trust is a subjective matter.

(c) **Objective basis** - Although the evaluation of trust itself is subjective, its sources which are later introduced as Trust Indicators (TIs) should be objective.

(d) **Automated management** - Trust needs to be modelled by a formal approach in order to be usable in VO computer systems.

(e) **Comparable** - Trust needs to be comparable among different organisations in order to model them within a common reputation system and support a fair decision process.

(f) **Dynamic** - Trust develops and changes over time.

2. A reputation system with above trust model as its core having the following properties:

(a) **Standardised access** - Functionality is offered to a trustor requesting a trustee's reputation by a reputation service built upon open standards.

(b) **Configurable** - Reputation evaluations in VOs from different classes also differ. The system can be configured to cater for the entire range of VO classes.

(c) **Extensible** - New trust sources or feedback may dynamically be added to be aggregated towards reputation.

3. An attack classification detailing possible attacks to subvert the reputation systems decision support and their mitigation.

4. A Multi-Agent Simulation framework to evaluate the reputation system in different simulation scenarios scenarios. Scenarios are defined to evaluate the system in VO application specific settings, perform a sensitivity analysis, analyse the system's resilience against specific attacks and further more.

While STORE claims to provide reputation based decision support for all classes of VOs, the evaluation can not take all of these classes into account. The VO classification [32], this thesis builds upon, reveals that VOs can not be distinctly separated, but span a continuous space of VO classes that frequently show overlapping properties. The evaluation is therefore conducted in the two VO classes at the far ends of this VO space, in ad-hoc service provisioning and collaborative engineering VOs.
While STORE's reputation model offers an integration of less sophisticated trust sources, e.g. transactional feedback provided by collaboration partners, the conceptual work and evaluation presented in this thesis focusses on the novel trust and reputation model, that is based on Trust Indicators and takes a stochastic approach.

The reputation system mentioned as the main contribution of this thesis is called STORE, STochastic REputation System. The following chapters detail that it meets all of the above trust requirements. Its design already makes it resilient against a large amount of known attack classes against reputation system. STORE's trust model roots an organisation's trustworthiness in observable, measurable properties characterising that organisation's trustworthiness. It does not exclusively rely on submitted feedback that is more susceptible to cheating and tampering as employed in attacks. STORE's evaluation shows that it very well distinguishes organisations according to their trustworthiness. Not only polarised trust levels, e.g. a very trustworthy organisations and its opposite, can be identified, but also subtle nuances in an organisation's trustworthiness can be detected. An organisation for example that delivers services in ad-hoc service provisioning as well as collaborative engineering VOs is in general not equally trustworthy within both VO contexts. Being overall considered a trustworthy organisation, deviations from service delivery deadlines, e.g. due to infrastructure problems at one of its locations, lowers its trust level within an ad-hoc service

provisioning application context. Due to the short lifetime of this VO, such delivery delays deviating from expected, reliable behaviour result in a greater reduction of trustworthiness than in a collaborative engineering VO that may still tolerate such delays due to its longer lifetime. STORE is able to reflect such behaviour in a VO application specific reputation rating. Another of STORE's key advantages is its ability to quickly reflect dynamic trust aspects. If an organisation's trustworthiness changes over time, STORE's reputation rating soon shows this change. STORE's sensitivity is determined by its configuration and can therefore be tailored for VO application scenarios where quicker or slower uptake of changes is beneficial.

1.2 Structure of the Thesis

This thesis is structured into five chapters. Figure 1.1 illustrates the chapter flow.

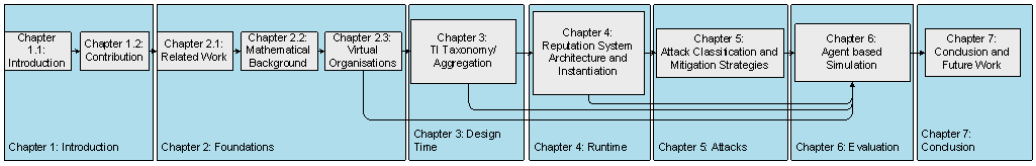

Fig. 1.1: Chapter Overview

This Chapter 1 started with an introduction into the topic of trust and reputation in Virtual Organisations. The contributions of this thesis were enumerated. Chapter 2 introduces the research foundation for this thesis. It first discusses the related work in the research fields of trust management and reputation systems. Selected closely related approaches are presented in more detail. The related work section finished with an analysis of research in the agent based simulation domain. Simulation approaches that were used in evaluations of reputation systems were considered especially relevant for this thesis. A set of mathematical background consisting of Bayes theory, Bayes networks, their application and evaluation, is provided next. After laying this groundwork, the foundations of VO research are presented. The main section in this chapter contains the analysis of the problem domain - partner selection in VOs. The problem definition departs from a VO classification and uses two motivational VO scenarios set in the collaborative engineering and ad-hoc service provisioning application domain. This section delivers the requirements for a reputation system that is able to provide automated decision support for VOs in general.
Chapter 3 begins documenting the main technical contribution of this thesis, the STORE reputation system. STORE is designed to meet the requirements from the previous chapter and provide automated reputation based decision support for the partner selection in VOs. Following a top-down approach, first the underlying trust model of trust sources, so-called Trust Indicators and their taxonomy is introduced. Followed by the detailed model of individual Trust Indicators, six indicator examples are introduced. Each of them is analysed with respect to its relevance for the previously described VO scenarios.
STORE's runtime aspects such as its architecture, deployment and bootstrapping process are discussed in Chapter 4. An example of a deployed STORE system is based on the six Trust Indicator examples from the previous chapter. It finishes with other non-functional system aspects such as information system and communication security requirements, a productive STORE system has to meet and thoughts about the ownership model.
Chapter 5 provides an attack classification on reputation systems in general and STORE in particular. For each class of attack, mitigating measures are suggested.

The following Chapter 6 evaluates STORE in different VO application scenarios adopting a multi agent based simulation methodology. In the beginning of this chapter, the simulation environment itself is introduced. This entails the sections about the agent model and the simulation environment's architecture. An emphasis, including examples, is put on the relationship of configurable agent classes and how these relate to Trust Indicators. In order to be able to compare STORE across different VO application scenarios, the concept of an agent's Quality of Service as the basis for a VO independent measure is introduced. In the second part of this chapter, STORE is evaluated within the two VO application scenarios from collaborative engineering and ad-hoc service provisioning. Further simulation scenarios compare STORE to a closely related reputation system also following a stochastic approach and how it fares against selected attack classes from the previous chapter.
Chapter 7 concludes this thesis and provides an outlook on future work.

2. FOUNDATIONS

This chapter bundles the scientific background for this thesis. It encompasses the related work of the research fields in trust and reputation management, related to the core contribution of this thesis, the STORE reputation system. Then, related work from the agent based simulation domain is discussed that constitutes the evaluation methodology chosen for this thesis. Afterwards, a brief, but essential section addresses STORE's mathematical background. This section is restricted to an introduction into Bayesian theory which motivates the adoption of this approach for a novel trust model. Further, more detailed, mathematical background addressing particular stochastic distributions and their properties which are used in the STORE model are outsourced into Appendix A.1. This chapter's main section contains the problem definition. Related work from VO research is presented, as well as a VO classification complemented with a detailed description of two concrete VO application scenarios which motivate the problem definition. The VO application scenarios originate from the opposing, outlying ends of the VO classification, collaborative engineering and ad-hoc service provisioning.

2.1 Related Work

The related work in this section is divided into sections dealing with different aspects and subsystems of the work conducted in this thesis. These are namely trust management and reputation systems followed by related work from the evaluation methodology, (Multi-) Agent Simulation.

2.1.1 Trust Management

As stated in the beginning, the notion of trust stems from human sociology, describing aspects of human relationships. Aspects encompass dependability, reliability in another human's behaviour, introducing situational and context properties. In [44], Gambetta explores trust from that angle and lays the foundation to use trust also in an IT environment. A small adaptation to fit it into a VO context leads to the following definition of trust for this thesis:

Definition 1: Trust is the subjective probability by which one organisation expects another to behave reliably, more concretely, the expectation in an organisation's ability to perform actions captured in a role specification within the context of a VO.

This definition details the most important properties of trust:

- Trust is subjective - the amount of trust in a trustee may vary among different trustors
- Trust is uncertain - in trusting someone to a certain extent, the trustor encounters the risk of trusting too much or little

- Trust expresses a belief or expectation of a trustee's reliability - this reliable behaviour is depending on a context, for instance given by the trustee's duty in a VO

Luhmann [75] follows emphasizing on a lack of ordering in trust aspects and an ubiquitous confusion about trust and related terms such as confidence, uncertainty, loyalty and familiarity. A certain level of uncertainty is required for trust to emerge. Relationships are forged based on confidence, while loyalty and familiarity are aspects of the subjectivity of trust. Luhmann also hinted at trust's relation to risk, being a solution for specific problems of risk. In his opinion, this can only be achieved in a world of familiarity hinting at required information, knowledge of previous interactions and the identity, about an entity such as a business partner.
Blaze, Feigenbaum and Lacy then define the term "trust management (problem)" as the collective study of security policies, security credentials and trust relationships [14]. This fundamental working definition regards trust from a purely technical perspective and provides a reduced angle on trust, suitable to work within an IT environment. They are the recognized inventors of the term "trust management" and also provide a trust management implementation called PolicyMaker, followed by KeyNote. PolicyMaker though primarily aims at providing access control in decentralized environments.
Patton and Jøsang, even though in the specific context of eCommerce, define in [79] Trust Management as

> ... the activity of collecting, codifying, analysing and evaluating evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding e-commerce transactions.

Traditionally, trust management approaches are based on a particular domain specific and authoritative Public Key Infrastructure (PKI). PKIs lead to a rather static trust model which fulfils e.g. security requirements for authentication or non-repudiation, but are less well suited for flexible, sophisticated trust management requirements, e.g. for reputation management in a dynamic VO. In particular, a PKI only supports binary trust decisions. Most PKI based approaches are based on X.509 [17] public key certificates that identify the principal owning the corresponding private key. A trustor receiving such a certificate can only decide to trust or distrust the trustee, finer grained degrees of trust can not be modelled relying only on a PKI. Exactly these finer grained degrees of trust are required for collaborating principals in VOs and similar organisational structures. Such degrees of trust allow to distinguish among principals that are more or less trustworthy in particular application scenarios. A more flexible and also realistic approach based on domain specific authorities still using PKI mechanisms is called SPKI/SDSI [35]. Realistic in a sense that in the real world no single trusted organisation exists serving as the root authority for all digital certificates. The world resembles more a set of domains having domain specific authorities, issuing authoritative statements, certificates only for their own domain. Trust links between domains are realized by means of cross-certification. This trust model is taken up as the Hybrid-PKI-Model and extended by different forms of mediation among entities in [13] and [68].

Since a global hierarchy as pursued in X.509 PKI models does not scale for most scenarios, SPKI/SDSI allows for multiple certification authorities each in one well defined domain. SPKI/SDSI provides a better matching model for real-life scenarios as in VOs. However, the decision making and aggregation processes delivering the foundation for the mediators main purpose are not a focus of this work.

Research taking a perspective on trust suitable for VOs, employing a stochastic approach or using higher order logics for reasoning is conducted for instance by Jøsang [79, 66]. This work is laying a foundation for providing trust controls to the BP layer. Jøsang first departed from his perception of reasoning about

trust, where he specified his own flavour of modal logic for reasoning support about trust, called subjective logic[66]. More recent work which is based on a particular distribution embedded in a reputation system is discussed in the following section.

Quite frequently in literature, trust and reputation are handled synonymously, an explicit clarification of the mutual relationship between both terms is missing. In [83], Mui et al. take the approach of inferring trust from reputation in a multi-agent network. Reputation is defined as an expectation of future agent behaviour based on probabilities. In [104], Sabater et al. define reputation as the "opinion or view of one (agent) about something". The basis for reputation are so-called impressions, a generalisation of trust indicators. This reveals another ambiguity between recommendation and reputation, orthogonal to trust and reputation. Frequently, reputation is seen as the aggregation of **subjective trust values or parameters** to an **objective trust measure** [2, 40, 30]. That view is also argued for by the author of this thesis, e.g. in [49].

A final cornerstone which is clearly visible in related work about trust establishment is the importance of previous experience, a record of prior-knowledge with an entity. Trust establishment without such information is challenging [2, 104, 14, 128, 49].

2.1.2 Reputation Systems

Reputation systems are typically used among members of a community to track their opinion of each other. A reputation system aggregates multiple subjective trust measures, striving to obtain a more objective reputation measure. The following definition sets the understanding of the term reputation for the remainder of this thesis.

Definition 2: Reputation is the objective, business context specific aggregation of trust values from multiple independent sources to support an organisation A's, the trustor's, decision making process with respect to an intended collaboration with another organisation B, the trustee.

Community members influence reputation of other entities by rating each other's performance according to their opinion and send queries for other member's reputation values. Two dominating classes of reputation system architectures emerged in literature, centralised and distributed ones. Following a centralised reputation system paradigm, one central system is available in a community. It represents a central point of contact that community members can send their queries to. Distributed reputation systems have no central system instance, each member maintains his own reputation database about others and implements common functionality to compute reputation values. A member send his queries about another community member to others in his neighbourhood and computes a reputation value based on the received answers. Centralised systems keep track of member opinions at a central spot with reduced communication overhead, but may be easier compromised and implement a single point of failure. Distributed systems are resilient against failure of individual community member functionality but require a lot of communication among members and each member is responsible for the reputation computation from a varying number of member opinions. Interesting communities from a business perspective are eMarketplaces[1], auction sites[2] exhibiting a transactional behaviour in their member's interactions. Reputation systems in such environments are used to tag misbehaving members and limit their negative behaviour's, e.g. fraudulent behaviour and cheating, impact on the community system. These commercial sites were analysed from a trust and reputation perspective by

[1] for instance Amazon, http://www.amazon.com
[2] eBay, http://www.ebay.com

Resnick et al. [98]. In [97], Resnick and others performed a controlled experiment on eBay to assess the impact of reputation on buyer behaviour, more specifically a buyer's willingness to pay more for a product from a highly reputable vendor. The experiment, conducted with postcards sales, was afterwards analysed in depth using stochastic models and methodologies. Other, non-commercial, communities relying on reputation systems for regulating misbehaviour are Slashdot[3] or Kuro5shin[4]. All enumerated communities are included in a more recent survey [65] published in 2004. The analysis was conducted focussing on accuracy and resilience of their underlying online reputation systems.
However, Bolton [15] also produced a survey investigating the effectiveness of online reputation systems. The results are not encouraging, productive systems already used at commercial sites are still vulnerable for fraudulent behaviour. Those results motivate the focus on rooting trust management in trust indicators, modelling the semantics of entities and their behaviour to allow in the end for accurate and precise reputation supported decisions. The initially claimed quality of this approach stems from exactly those verifiable evaluation criteria, accuracy of predicted parameter development and precision of the model compared to observed, realistic behaviour and development. Dellaroca[28] analysed attacks and threats to reputation systems in online communities with buyers and sellers, classified threats according to their properties and suggested countermeasures. It showed that attacks are possible against buyers and sellers ("badmouthing", "discrimination" or "ballot stuffing"), posing different degrees of difficulty to the reputation system to counteract. Generally, attacks against buyers are easier to counteract than attacks against sellers since the crucial point is identity management. Community platforms tend to implement a sound identity management for sellers, e.g. through a registration process with subsequent identity verification. Since such platforms use heterogeneous mechanisms ranging from anonymous usage over pseudonyms to a sound identity management on the buyer side, buyer identities are easier to conceal than seller identities. Dellaroca argues with statistical methods, but he only focuses on the ratings over time which he assumes to be normally distributed. The mentioned threats, reputation systems are susceptible to, cheating and other fraudulent behaviour suggest to have a closer look at how to mitigate those. Padovan et al. [16] address the issues of liars and stability in the AVALANCHE reputation system. To achieve their goal of reducing complexity in high risk ad-hoc cooperations among strangers, they integrate technical and non-technical security into their system. But, with the exception of initial trust values, they see the reputation system itself as a means to counteract fraudulent behaviour in collaborations/cooperation rather addressing reputation system internal fraudulent behaviour. In [107], Sen and Sajja address the problem of liars in a community of agents with varying environmental constraints, e.g. percentage of liars and performance of agents. They provide a decision mechanism to allow the identification of a lying agent with a certain probability to increase the robustness of their reputation system.

More concretely, the following published reputation systems, most importantly KeyNote, Sporas/Histos and the beta reputation system, are considered closely related work.

Related to the idea of modelling characterising properties as trust indicators, Xiong and Liu introduced the term trust parameter in [128]. They introduced a reputation based trust model called PeerTrust for eCommerce communities. They state the goal of a predictive trust model, allowing to foresee future peer behaviour. But instead of truly modelling characteristics of, in their terms, peers which relate to organisations in this work's context, they simply extend subjective feedback through rating. The extension comprises of an absolute amount of successful transactions previously conducted with a certain peer and a community specific reputation value for that peer. So far, the related work suggests that there is a need to root trust in an entity itself, e.g. in its characterising properties or parameters. Since trust unites a multitude of dimensions,

[3] http://www.slashdot.org
[4] http://www.kuro5hin.org

social, economical, operational, financial and many more, this requires possibly a large amount of parameters to be considered for trust management. To efficiently deal with such an approach, the parameters have to be efficiently modelled and aggregated. An aggregation already hints at a classification of parameters delivering the root of trust. In [119], Viljanen did an effort to survey existing trust management approaches such as KeyNote [14], PolicyMaker, Referee, Poblano etc. and came up with a coarse grained classification of aspects which contribute to trust models. The models were then assessed in terms of their identity, action, business value, capability, competence, confidence, context, history and 3rd party awareness. This classification referred to the trust models as a whole, not specific parameters and was used to form an ontology of trust. A similar approach resulting in a so-called Trust Matrix was taken by Tan in [112]. The first matrix dimension if formed by similar trust model aspects as in [119], the second dimension is formed by a trust classification into party trust, control trust , potential gain and risk. The four trust classes all contribute to the notion of transactional trust, the centrepiece of the trust matrix.

Trust indicators are dynamic over time, especially in a VO environment, organisations evolve and change, so do their characterising properties. An usually underestimated task is the update of trust indicator models reflecting those changes. This requirement was frequently stated in a generic fashion, also specifically for trust indicators by Ruohomaa et al. in [103].

Looking at the aggregation of subjective values to an objective trust value, reputation, the following work is considered related. Sporas and Histos [129] together form a sophisticated reputation system for online marketplaces. Sporas, the reputation mechanism is used to bootstrap the web or graph based Histos. The latter, Histos, is more interesting in this context since it applies a certain probabilistic background. The underlying function 2.1 can be interpreted as an approximation of a stochastic density function based on a relative frequency of trust values. The reputation calculation based on the following formulae takes previous transactions into account:

$$R_{t+1} = \frac{1}{\Theta} \sum_{i=1}^{t} (\Phi(R_i)) R_{i+1}^{other} (W_{i+1} - E(W_{i+1}))) \tag{2.1}$$

$$\Phi(R) = 1 - \frac{1}{1 + e^{\frac{-(R-D)}{\sigma}}} \text{ and } E(W_{t+1}) = \frac{R_t}{D} \tag{2.2}$$

Where

- t is the number of ratings the user has received so far
- Θ is a constant integer greater than 1
- W_i represents the rating given by the user i
- R^{other} is the reputation value of the user giving the rating
- D is the range of the reputation values
- σ is the acceleration factor of the damping function Φ

It hereby can be observed that the smaller the value of σ, the steeper the damping factor $\Phi(R)$ which triggers the system ability to ”forget”.
Histos, the more complex reputation system uses Sporas to bootstrap and creates a dense web of pairwise ratings which can be visualised in an interaction graph, weight edges denoting the ratings.

In a more recent work, Regan et al. [94] provide a high level classification of reputation systems for electronic market places. They refer to global, personal and social classes of reputation systems. In the context of this work, such distinctions are more rooted in the system's, marketplace or VO, trust model than differences in the reputation system's architecture. On a technical level, e.g. a global and a personal reputation system may be the same, if such a system is hosted by a Trusted Third Party (TTP) instead of an entity querying for reputation. In that case, a TTP hosted reputation system would be characterised as a global reputation system in Regan's work. Reputation is based on information, e.g. ratings, with an inherent uncertainty. The authors focus on epistemic uncertainty resulting from a lack of knowledge about non-deterministic system behaviour, as in Equation (2.1). The core of their work under the umbrella of a social reputation system takes a Bayesian approach. Focussing on binary events modelled by Markov Decision Processes (MDPs), they are able in a sense to aggregate reputation using stochastic system theory, but the reputation calculation itself is rather limited in quality compared to real application scenarios due to the source in binary events. Moreover, the aggregation with MDPs follows a deterministic state model only partly transitioned by a stochastically defined belief function based on explicitly received certainty factor's queried and disclosed from other entities. This leads to a belief function which delivers actually a probability distribution over states. In the end, the system called the Advisor-POMDP based on Partially Observable Markov Decision Processes is able to find optimal MDP strategies for binary events in electronic marketplaces between the roles of buyers and sellers, but is not sufficiently flexible and does not deliver the quality of reputation allowing for predictions in VO environments. Furthermore, the reputation calculation is only based on a form of weighted ratings from buyers for sellers.

The Beta distribution is a center piece of Jøsang's more recent work and the core reputation assembler of his Beta reputation system [62]. Using an actual distribution is similar to the work in this thesis, but restricted to the Beta distribution which is well suited to model binary events, it is not possible to realistically model the true heterogeneous behaviours of trust indicators in VO environments with the required quality. The Beta reputation system is ground breaking since it introduced a stochastic distribution to implement its trust model, but still is limited to series of binary events, e.g. modelling transactions with satisfactory or unsatisfactory outcome. The Beta distribution is based on the Gamma function and defined as:

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{2.3}$$

Where

- p is the events probability (for satisfactory outcome)
- α denotes the observations with satisfactory outcome
- β denotes the observations with unsatisfactory outcome

If r observations with satisfactory outcome were observed and s with unsatisfactory, to calculate the probability based on those observations, simply α has to be set $r + 1$ and β $s + 1$ for the next occurring event in the series. r and s are for instance feedback ratings of an entity. Having two entities enacting for instance the roles of buyer (r_1, s_1) and seller (r_2, s_2), the collaboration between both is modelled with the joint distribution parametrized by $r = r_1 + r_2$ and $s = s_1 + s_2$. Stochastic dependencies between both roles, e.g. reflecting contractual dependencies, are ignored.

In contrast to (stochastic) process theory, Bayes theory is frequently employed to aggregate probabilities or model a network of interacting entities, the latter with a so-called Bayes network. In [88], Nurmi uses

Bayes theory in the form of conditional probabilities in a game-theoretic reputation model. Based on simulation, he compares it to eBay and can show that the number of successful transactions increases significantly in his simulation framework. Frequently, in related work such as [120], collaborating entities or agents are modelled as nodes in a structured directed (acyclic) graph. Trust is based on usually simple parameters, e.g. response time, denoting entity properties and calculated as a (vector of) conditional probabilities. Interactions among entities can then be assessed by walking the graph, comparing all conditional probabilities of passed nodes and taking optional given edge weights into account. The path with the highest probability for a successful interaction is also the most trusted path and the preferred interaction partner can be chosen. The drawback of such a simple, purely Bayesian network based approach is the lack in scalability. Theoretically, a change in one parameter requires the recalculation of all conditional probabilities, a complete network update. This is expensive for large networks[5], the tradeoff when making localization assumptions, is a delayed parameter update propagation through the network.

In the reputation system literature, a gap in the underlying trust model becomes apparent. Reputation systems such as for instance PeerTrust [128] introduce a trust metric which serves the trust values or later reputation building. Depending on the trust metric's level of sophistication and complexity, different facets of trust may be addressed such as entity relationship history, possibly appearing as a transaction history in eCommerce environments, and related community context and transaction satisfaction levels. Such trust models evaluate their own performance by relying on explicit feedback from participating entities, e.g. as provided ratings or complaints. This information is then used to evaluate the trust model explicitly and to counteract above attacks. But such explicit measures of evaluation are not enough since in such work, a game theoretical reliable behaviour of participating entities is frequently assumed. This means, an entity is either consistently behaving bad, delivering wrong ratings or consistently good which in turn influences a community measurable factor, e.g. a credibility factor. Furthermore Ashri et al. follow in [10] a relationship based approach. They state similarly that relying simply on explicit feedback such as ratings does only transform the trust problem from the transaction partner to the provider of the rating. But they only follow with their relationship based approach, taking e.g. the entity's organizational structures into account, another path using explicit feedback. An implicit trust model evaluation which does not rely on external ratings, which allows evaluation of a trust model instance only based upon model characteristics would increase the reputation system's resilience against attacks. In a previous work [104], one of the authors, Jordi Sabater, presented the REGRET reputation system which calculates reputation as a subjective agent property aggregating a set of prior interaction history with other agents, simply called "impressions", by a general mean. This approach suffers of a lack of flexibility in modelling the trust roots and also their aggregation but a time variant function was introduced in the evaluation of the generalised mean, giving more recent impressions a higher weight.

VOs represent a communication and organisational community model in between a fully distributed and centralised network topology. Homogeneous VO member domains are temporarily joined to a heterogeneous, collaborative VO structure. Reputation systems in fully distributed environments have to deal with the Byzantine problem [6] introducing complexity and overhead in handling reputation values. A fully centralized system needs less communication effort but requires the services of a trusted, maintaining party, predominantly a TTP. For VOs, a centralised reputation system architecture introduces more advantages than drawbacks. If a TTP among the VO members can be agreed upon, an external party can be nominated to host and own the reputation system. This may give rise to a new business model offering services to

[5] VOs, especially in the collaborative engineering domain, can reach a considerable size depending on the amount of specialised suppliers needed to manufacture a complex design

[6] an agreement on an initially true value by n parties with m corrupted ones which has to validate and terminate

VO ecosystems. A reputation system owner may specialise on answer reputation queries about community members in dedicated application and VO domains. Since the STORE approach is not based on subjective opinion e.g. as feedback, but on observable TIs, the questions remain how these observations can be obtained by a centralised reputation system from sovereign VO member domains. If these observations for confidential TIs are not willingly provided or can not due to legal reasons, a semi-centralised approach can be taken. Confidential TI data can be pre-aggregated or anonymised in a VO member's domain and then submitted to the reputation system. More details and other approaches for specialised VO environments are discussed in section 4.2. VOs are supposed to be formed and become operational quickly in order to address an emerging business opportunity. This speed requirement leads to an on-demand creation of VOs, especially in more dynamic business domains. In order to provide automated reputation based decision support for such on demand created VOs, a reputation system must be always available. Therefore a centralised online reputation system appears to be the best choice for VOs.

The following Table 2.1.2 lists the most important trust properties of related reputation system that can be compared with the STORE approach and its trust requirements as stated in Section 1.1. These are namely how trust properties are classified, how the trust relationships are bootstrapped when a new reputation system is deployed, how the reputation context is captured, how and on which basis reputation measures are computed and how the dynamic trust aspects are taken into account. It becomes apparent that STORE fares especially well with respect to the required fine grained degrees of trust that are required to assess a principal's trustworthiness within an (VO) application domain. This goal is also supported with one of the thesis' core contributions, the TI taxonomy. Trust is not only based on a singular, uniform and subjective source such as feedback, but a set of sources that inherently describe an organisation's trustworthiness are selected to form the basis of STORE's trust model. Another advantage of STORE's approach manifests when dealing with dynamic aspects of trust. Changes of a principal's trustworthiness over time are resolved and taken into account by STORE's trust model since TIs are observed in regular time intervals.

2.1.3 (Multi-) Agent Simulation

Using a (Multi) Agent based Simulation ((M)AS) methodology to evaluate reputation systems is well established in literature. Mui et al. provide a comprehensive review of such usage in [84]. Since, in general, reputation systems offer decision support to a community of interacting actors, MAS is a suitable evaluation methodology of choice due to the following central facts. Community members, especially so in VOs consisting of otherwise sovereign organisations, can be well represented by autonomous intelligent agents. Important properties of such agents are the abilities to

- interact with their environment, including each other
- autonomous action
- (proactively) react on other agents
- socially comprehend and communicate
- act rationally and adapt

These properties allow a realistic model of an environment for decision support processes [39, 118, 25]. MAS allow a flexible role assignment to (groups of) agents and scale well in environments of varying amounts of agents. These properties are prerequisites for evaluating STORE in a simulated VO environment which is similar as to simulate market environments such as [108, 81].

Trust Properties/ Reputation System	PKI	Amazon/ eBay	Beta Reputation	Advisor-POMPD	Sporas/Histos	STORE
Trust Property Classification	certificate properties (mostly static property hierarchy)	arbitrary trust levels (transaction feedback)	binary event (transaction feedback)	binary event (transaction feedback)	binary event (transaction feedback)	TI Taxonomy and domain specific trust levels
Bootstrapping	explicitly trusted root CA	static initial value assignment	uniform distribution (uninformed choice)	static initial value assignment	static initial value assignment with SPORAS function	initial stochastic TI model ((uninformed choice) or TI data from a similar VO
Reputation context	none	transaction outcome, arbitrary levels	binary transaction outcome	binary transaction outcome	binary transaction outcome	implicit in TI model and requester sets VO specific emphasis on trust classes
Aggregation	n/a	arithmetic calculation	beta distribution	Markov process	Histos function	TI specific Bayesian distribution assumption and Bayes network
Dynamic trust aspects	infrequent issuing of new certificates	voluntary rating	voluntary rating	voluntary rating, deterministic state model	rating and forgetting over time	regular mandatory TI observations in TI specific time periods, optionally rating

Tab. 2.1: Comparing related work

A large class of reputation systems that base their calculation on transaction feedback have a simple agent model in common. Jøsang's Beta Reputation System was also evaluated in a MAS [64, 67] that simulated an eCommerce market environment. Agents modelled for this class of systems are commonly configured for a very simple behaviour, for instance being binarily honest or dishonest. They stick to this behaviour statically throughout the simulation. The MAS required to evaluate the STORE reputation system can not only rely on transaction feedback, but must take the rich underlying trust model, the TIs, into account. An agent and its behaviour during the simulation is defined by its TI data which in turn is based on periodically updated probability distributions. In TRAVOS (Trust and Reputation model for Agent-based Virtual OrganisationS) [113], Teacy et al. aimed at modelling a more realistic agent behaviour by introducing a third agent class that behaved honestly but its feedback was distorted with stochastic noise. Therefore, the agent could statistically behave dishonestly. But still, this behaviour was configured and the agent was not able to behave radically different, e.g. switch classes, throughout the simulation. Most MAS take turns, agents playing simple buyer/supplier or similar roles interact in rounds. Agents are typically matched in pairs and their interaction is reduced to one atomic transaction. But agents, depending on their configured behaviour frequently interact along game theoretic strategies. In many MAS, agents play the Prisoner's Dilemma from Axelrod et al. [99]. A MAS to evaluate STORE is required to take, in general, n agent classes into account, one agent class per VO class that distinguishes between application specific VOs. Agents are required to be matched asymmetrically, one VOM selects usually more than one VO member.

Schlosser et al. developed a agent based simulation framework [105] specifically to evaluate and to compare reputation and recommender systems. The framework takes care of agent model and maintenance, as well as the overall simulation control. A reputation system designer only has to implement his reputation mechanism in the framework following a well defined interface. The framework was tested with simple reputation mechanisms from eBay, Amazon and more advanced such as the Beta Reputation System. The framework is unfortunately not able to deal with STORE's rich trust model and stochastic aggregation mechanism. The agent behaviour is also configured statically again. The Agent Reputation and Trust (ART) testbed initiative, partly from the same authors, [42] aimed at establishing a testbed for agent trust- and reputation-related technologies. This testbed serves two roles: (1) as a competition forum in which researchers can compare their technologies against objective metrics, and (2) as a suite of tools with flexible parameters, allowing researchers to perform customizable, easily-repeatable experiments. The ART testbed is also not suitable to evaluate STORE due to the same constraints as the framework from Schlosser et al.

In [100], agents represent companies/firms and not necessarily individuals. A market is analysed but from an economic perspective. Turnover, investments and other economic indicators/inputs are used to validate the author's claims about reputation impact on the market by mathematical proofs, calculated in closed form. More concretely, the author's proof their claims by stating that reputation leads to stable (or unstable) equilibria which are desired (or not).

Other reputation system evaluation methodologies besides MAS in literature are for instance case studies. Resnick et al. were able to obtain anonymised transactional data from eBay. In [95] they performed an analysis of the eBay reputation system for the case of a constraint user population of card collectors, identifying several possibilities to cheat giving false feedback on eBay. Other approaches favour empirical setting with human players in a trading market with virtual money. Nurmi evaluates his Bayesian reputation system in [88] which humans whose trade history has a similar structure as the transactional data in the previous eBay case study. These approaches have the disadvantage that they can be only valid for one VO application scenario per study or experiment. Due to its flexibility in configuring application specific settings and agent populations, a MAS approach is better suited to evaluate STORE's capability to provide automated decision

support for VOs in general.

Studying related work in evaluations of reputation systems revealed MAS as the best methodology to evaluate STORE. Existing MAS models and frameworks fall short in coping with the trust model that underlies the STORE design. A MAS framework specifically to deal with the STORE model needs to be designed for its evaluation, meeting the following requirements:

- The MAS agent model must allow to define n agent classes, one class per VO application or industry, with different properties
- The MAS agent model must allow to configure dynamically changing agent behaviour
- The MAS model must allow VO application scenario specific settings, e.g. for agent population, lifetime or economic parameters
- The MAS agent model must allow to define VO application scenario specific preferences for the underlying TI based trust model
- The MAS model must allow the comparison of simulation results across VO application scenarios

2.2 Mathematical Background - Bayes Theory and Stochastic Aggregation Techniques

This second foundational section motivates and describes the mathematical background used as aggregation approach in the STORE reputation system. It consists of mathematical background about Bayes Networks and related theorems used for stochastic TI aggregation. Further mathematical background such as an enumeration of density functions serving as distribution assumptions in the TI models is presented in Appendix A.1.

The Bayes theory was founded by Reverend Thomas Bayes (1702-1761), a British mathematician. It addresses the problem of ”inverse probability” by introducing conditional probabilities. The Bayes approach was chosen for the line of work in this thesis due to the substantial overlap between trust and the aspects addressed by the Bayes theory, namely:

- The input information for Bayes models is usually of a subjective nature - as is trust.
- Updating information (with evidence from defined sources) relies on Bayes conditioning as its basis which relates to the requirement for a sound trust model.
- Bayes theory distinguishes between causal and evidential reasoning which fits the STORE approach, assuming objective trust measures as TI evidence updates and the aggregated reputation value that is inferred as the product of causal reasoning.

Especially the last aspect was only posthumously published by Thomas Bayes in 1763[12].

The Bayes Theorem is one of the most important building blocks of his work which is also used in this thesis. In general, it relates the conditional and marginal probabilities of two random events. In particular, it is used to calculate a posterior probability for a random event with given observations.

In general, the Bayes theorem is often given in the following formal notation:

$$P(\Theta|X) = \frac{P(\Theta)P(X|\Theta)}{P(X)} \tag{2.4}$$

Alternative forms exists as well. The following two representations relate to the use of random variables with first continuous and then discrete distribution assumptions. Evaluating the variable's partitioning leads in the first case to an integration, in second to a summation of the respective partitions.

$$P(\Theta|X) = \frac{P(\Theta)P(X|\Theta)}{\int_{-\infty}^{\infty} P(\Theta)P(X|\Theta)d\Theta} \tag{2.5}$$

$$P(\Theta_i|X_j) = \frac{P(\Theta_i)P(X_j|\Theta_i)}{\sum_{k=1}^{N} P(\Theta_k)P(X_j|\Theta_k)P(\Theta_k)} \tag{2.6}$$

In all three representations, the variables carry the following meaning:

- $P(\Theta)$ is called the prior (or also marginal) probability of Θ. Prior hereby denotes that X has no influence over Θ.
- $P(\Theta|X)$ is called the posterior (or conditional) probability since it depends on X.
- $P(X|\Theta)$ is called the likelihood probability which is also a conditional probability depending on Θ.
- $P(X)$ is called the prior or marginal probability of X and serves as a normalising constant in the term's denominator.

Departing from this foundational work, probabilistic models based on Directed Acyclic Graphs (DAGs) started in many research fields. In the 1970s, such models were employed for uncertain reasoning in the Artificial Intelligence (AI) domain [89] and finally the term Bayes Network for a special variant of such models emerged [89, 90].
A *Bayes Network* (BN) can be graphically represented as a DAG. It consists of nodes which hold random variables and edges which denote causal relationships encoded as conditional probabilities between the random variables. BNs represent causal relationships of a human's understanding of the world. Random variables in a BN can either relate to directly observable events e.g. the occurrence of a phone call or a delivery of a good, so called *information or evidence nodes*, or propagate information through the network by evaluating the conditional probabilities along the edges, so called *inference or prediction nodes*.

Conditional probabilities are evaluated according to the Bayes Theorem 2.4. Each node depends on his parent(s). For instance an edge from node A to B makes A B's *parent* while in turn B is A's *child*. This leads to the general formula for evaluating conditional probabilities in a BN where the set of parents of a node X_i is denoted by $parents(X_i)$. In a BN consisting of n nodes, the joint distribution of the node values is defined as the product of each node's individual distribution stochastically depending on its parents.

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|parents(X_i)) \tag{2.7}$$

Whenever events related to variables of interest in information nodes deliver freshly observed data, the model requires an update. In the case of a BN, this requires a full net update, an evaluation of all conditional probabilities and random variables encoded within the net. For a large scale network, e.g. modelling a supply chain from the automotive domain with a couple of hundred nodes, this becomes a computationally expensive task following a naive approach e.g. sequentially calculating the products in an arbitrary order. Many advances in efficient evaluation methodologies for BNs were made in recent years. It is possible, by selecting an application specific node order that maximises the amount of internal random variables which are not visible beyond inference nodes, to greatly reduce the amount of required multiplications in each

update by variable elimination [72]. Further advances were made using localisation assumptions that partition the net and allow ”lazy” updates of outer lying regions and graph transformations using junction trees [115, 77, 78]. BNs are able to learn in two ways, first by parameter, second by structure learning. Parameter learning means providing new evidence as input for the BN whenever it becomes available. The work in this thesis focusses on this learning mode. Freshly observed TI data for instance are submitted into the BN's information nodes triggering updates for random variables in all other nodes. The second mode of learning, structural learning, requires altering the BN's structure and topology when it becomes evident that one structural version models the world more closely than the previous one. This mode of learning is very resource consuming and less well understood in Bayesian theory research than parameter learning.

The following example in Figure 2.1 illustrates the use of a BN model. Random variable X encodes the probability of a customer complaint, e.g. a phone call of a worried customer in a call centre because his new bought product shows a defect. Variable Y encodes the probability of a customer error, e.g. a customer wrongly uses his product which is not defective. X and Y are random variables in information nodes since they model observable events. Random variable D encodes the probability of said product to be actually defective. D is conditionally dependent on X and Y. Finally, R denotes the probability of a required product return and depends only on D, the probability of the product being actually defective. The latter expresses an expert's perspective of the world, explicitly modelling R as not directly dependent on X and Y. The node R is an inference node that would be queried for predictions if a product, upon complaints, needs to be returned. It also becomes evident, that setting the initial probabilities for the information nodes' events is crucial for the usefulness of the model and requires a skilled and knowledgeable domain expert.

These paragraphs introducing Bayes theory are sufficient to follow the remainder of this thesis. For the interested reader, the following publication, especially from Finn Jensen, can be recommended for more in depth material and applications [63]. One of the main motivations for choosing BN to serve as the aggregation methodology is the work of J. Pearl. In [90], Pearl et al analysed previous and current applications of BNs and closely related network structures. In all cases and applications, BNs show an advanced expressiveness in modelling the reality which is a distinct advantage over similar approaches such as stochastic processes. The STORE reputation system roots trust in a variety of heterogeneous TIs. Modelling these TIs with individual attributes such as domain, unit and distribution assumption, benefits from such advanced expressiveness. Norman Fenton et al applied Bayes modelling to risk management approaches especially within the software safety domain, providing interesting insight in the applicability and the problems a Bayes theory practitioner faces, also on a larger scale with respect to the size of the modelled network [86, 86]. Since choosing BN models means obviously taking a Bayesian's point of view, Fenton faced, among other problems, the trade off between a realistic model, taking as many interdependencies between events and their random variables as possible into account, and the hereby created uncertainty while evaluating an increased number of conditional probabilities leading to an increased variance in the random variables which are modelled to infer from the network. The same trade off is very well described and analysed in the work of David Heckerman et al [55] using decision networks. Decision networks are closely related to BNs since they can be roughly categorised as undirected BNs which inherently allow for cycles in the graph[54]. The example used for evaluation in their work was a graph generated from visitors and resources of several Microsoft web sites such as ms.com or msnbc.com. The goal was to visualise web site visitors and their preferred resources having predictions of future visitor behaviour in mind. The, for this thesis, most interesting conclusion they drew from their evaluation results was that it is, in many cases, beneficial for obtaining more to the point and certain inference results to limit the explicitly modelled interdependencies to the most important once. Such a modelling best practice avoids an ”overloading” of the BN model.

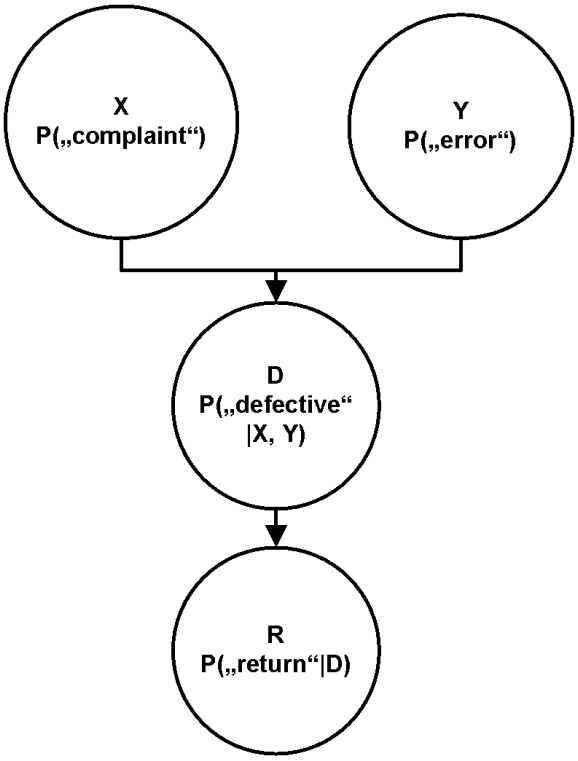

Fig. 2.1: BN Example: Customer Complaint Process

2.3 Virtual Organisations

Virtual Organisations (VOs) are proposed as the future of collaborative business systems especially in Business-to-Business (B2B) scenarios. VO proponents suggest that there will be key advantages for production and profitability realized, when short-term, specially contracted and objective-oriented cross-domain relationships are formed, without extensive geographic dislocation of physical resources and people [82]. Application scenarios for VOs, especially from collaborative engineering, however, suggest that such geographical dislocations are rather the norm. The preferred definition of a VO for this thesis was adopted from [27, 46, 80, 110]:

Definition 3: A VO is defined as a temporary alliance of otherwise independent, geographically dispersed organisations or individuals which pool their resources to achieve a business goal one alone can not master.

Such a business goal is typically identified by a dedicated organisation, e.g. a system's integrator in the aerospace industry learning about a government issued plane manufacturing tender. This organisation then starts identifying others bringing in specialised expertise to achieve this goal and selects a subset of the identified organisations to form a consortium. This coarse grained distinction gives rise to the assignment of distinct roles. In the following, the first, dedicated organisation is assigned the role of a *VO Manager* (VOM) who identifies and selects organisations of the role *VO Member* (M) [101].

Wolters and Hoogeweegen provide a comprehensive approach for a definition of a VO, other authors vary in their definitions but there are some recurring properties across publications [127]. They identify these as:

> Temporary alignments of a network of independent organizations, dynamic switching between network partners, end-customer requirements as starting point, bringing together the core competencies of the partners and intensive use of ICT (Information and Communication Technologies).

According to [110], a VO follows a phased lifecycle, consisting of five phases with varying requirements on trust and reputation management:

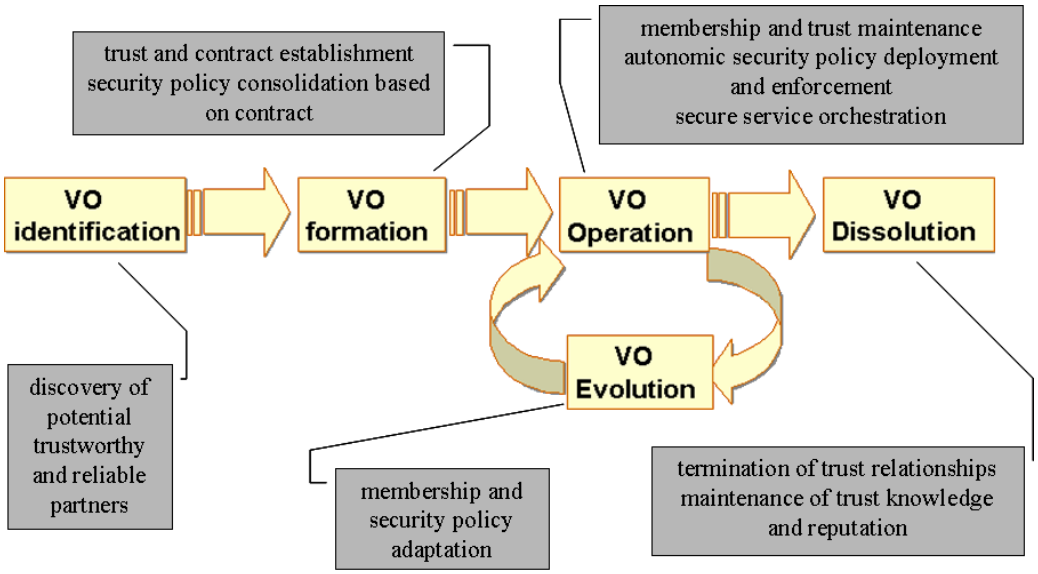

Fig. 2.2: VO lifecycle phases

1. **Identification Phase** - Potential VO partners are discovered, their trustworthiness is not yet ascertained.

2. **Formation Phase** - The most trustworthy VO partners are selected from the set of potentials for each required role. Contracts are established preparing for a collaboration on business level. The ICT infrastructure integration of each partner is prepared to allow for a seamless collaboration even on the IT level.

3. **Operation Phase** - The phase where the VO's actual work is conducted, BPs are enacted and the orchestration of business specific domain services takes place.

4. **Evolution Phase** - In case of unplanned events or accidents, operations have to be adapted, e.g. by replacing a VO partner of decreased trustworthiness by entering an Evolution Phase.

5. **Dissolution Phase** - Having met or failed the intended business objective, assets are dispersed according to the contractual agreements, trust relationships are terminated and trust information is maintained in this final phase.

Trust establishment becomes most crucial during the VO's formation phase. Trust and reputation management can support the VO's decision making processes with mechanisms which are designed to identify the most trustworthy VO members. The VO's operational phase denotes the time period, when the actual work to address the common business opportunity is conducted. The (ICT supported) work delivers a wealth of objective, observable data that characterises a VO member's trustworthy behaviour and can be used to predict that member's expected trustworthy behaviour in future VOs. Such data may be observed, for instance, from a VO member's operational sales and delivery processes or his financial development. Finally, the dissolution phase may be the source of a more subjective measure for a VO member's trustworthy behaviour. Organisations that transacted in the same VO may submit feedback about the other members they directly interacted with. Such feedback further improves predictions about the subject's expected future trustworthy behaviour.

2.3.1 Problem Definition and VO Application Scenarios

Having introduced VOs and their related work in Section 2.3 in general, this section is intended to delve into the details of particular VOs that are relevant for the further continuation of this thesis. The goal of the STORE reputation system is to provide automated trust based decision support for a wide variety of VOs in general and not only selected ones e.g. from particular application domains. To meet this goal, this section picks up the work from a previously conducted VO classification effort [32] and highlight the two contrasting VO classes opposing each other at the two ends of a continuous strand of different VO classes. Detailed application scenario descriptions are provided for these VO classes which serve as the base simulation scenarios for STORE's evaluation in Chapter 6.

VOs and VO like structures such as joint ventures can be observed in increasing number in the business world. Organisations and individuals choosing collaboration over fierce competition as their dominating business strategy are rewarded with higher profits and will be able to take more and larger business opportunities in the future. As mentioned in the beginning, the VO's business goal is posing an increasing demand at the formation speed. To achieve this requirement, organisations participating in VOs and similar organisational structures face the automation demand through increased reliance on and integration of Internet and Communications Technology (ICT) infrastructures. ICT integration does not only mean defining endpoints and message formats between collaborating enterprises and securing these with traditional hard security measures on infrastructure level such as Virtual Private Networks (VPNs), authentication and access control mechanisms. It also entails Business Process (BP) integration that a collaborative BP can be jointly enacted following an agreed upon choreography. Keeping that in mind, one is not surprised that Gartner estimated in 2002 the cost of a typical ICT integration endeavour in a VO like structure to eat up 30-40% of the company's IT budget. At the same time Forrester estimated a more conservative 30%. Pinning absolute numbers to the next described collaborative engineering VO example, an organisation's integration cost, deduced from its IT budget, amounts to M€15-20.

The speed requirement can be especially observed in dynamic application domains such as the high tech industry in collaborative engineering. Taking the example of chip manufacturing, it is evident that a large set of specialised manufacturers and suppliers contribute, under the umbrella of an integrator or VOM, e.g. Intel, to the common goal - design, manufacturing and shipping of a new chip generation. The high tech industry

is characterised by a high fluctuation of specialised suppliers, unsuccessful disappear and new ones emerge, as well as daily prices for components. These two reasons pressure integrators to frequently re-assess their choice in suppliers and select new, cheaper ones. Due to the high fluctuation, such a selection can not be based on an own, long established history with a particular supplier, but asks for other forms of decision support in selecting the right, most trustworthy business partner. Since most business processes are already heavily ICT based, a form of automated decision support e.g. from a reputation system can help in quickly selecting the most trustworthy business partner. Taking a closer look at Intel's chip manufacturing process, the need for swift and frequent partner selection processes becomes already apparent in the assembly of Central Processing Unit (CPU) heat sinks. The following Figure 2.3 illustrates the increase of specialised heat sink suppliers over the product lifecycle of three major CPU versions.

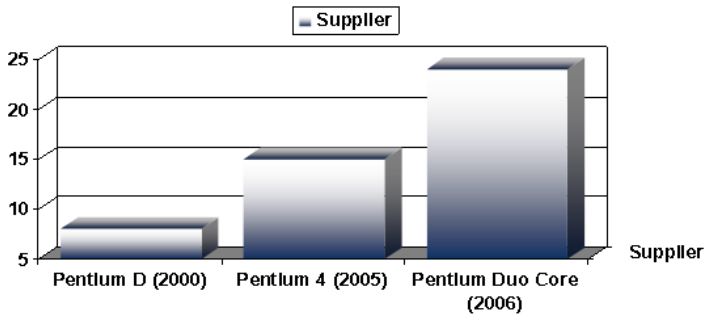

Fig. 2.3: Intel CPU - Heatsink manufacturers (Source: http//www.intel.com/design)

In 2000, 8 heat sink suppliers delivered parts for the boxed version of the Pentium D to Intel. The number increased to 15 in 2005 for the Pentium 4 and to 26 in 2006 with only one year in between due to a further speed up in the high tech industry's innovation lifecycle. Double of individual suppliers were already eliminated in this example. In conclusion, faster product lifecycles and increased numbers of specialised suppliers demand for trust based automated decision support in VOs [52].

Many European Union (EU) funded research projects such as "TrustCoM"[116] researched VO structures. In [32], a VO classification effort was undertaken, analysing VOs with respect to business, lifecycle as well as security, trust and contract management criteria. The author of this thesis contributed substantially to the classification effort. A continuous set of VO classes was identified, blending seamlessly into each other. The following Figure 2.4 provides a condensed overview.

It could be observed, that VO classes differ in a set of criteria, the ones relevant for the remainder of this thesis are stated in Table 2.2.

The resulting classification revealed a wide, overlapping range of VO classes from different business domains. The entire range is spanned by two contrasting VO classes:

- **Class I** - Rather stable and long-lived VOs are put into this class lasting for up to several decades. Collaborative Engineering (CE) is a typical business domain including comparably long-lived aerospace

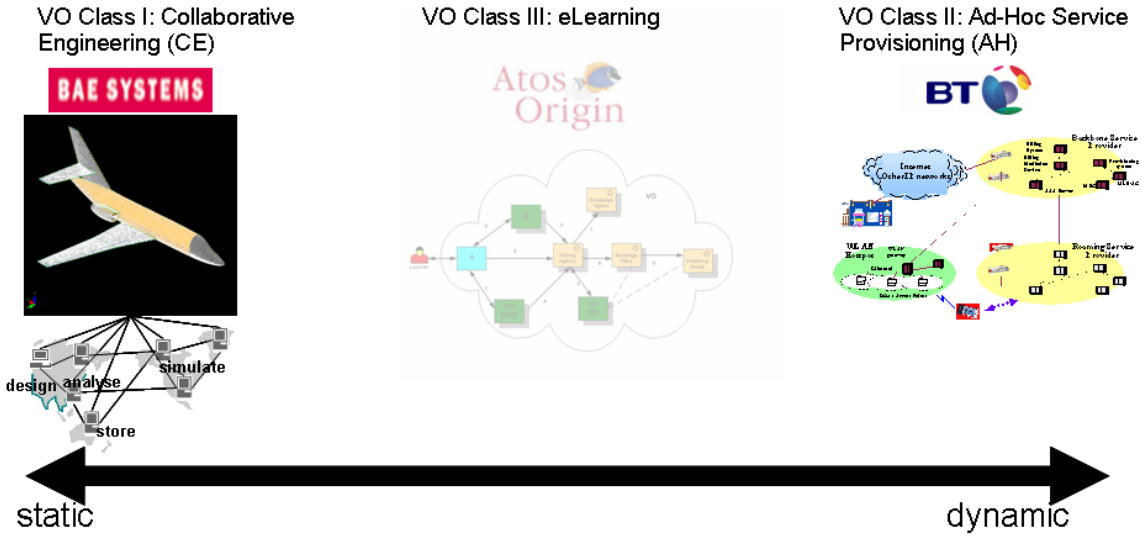

Fig. 2.4: VO classification overview

Criteria	VO Class I	VO Class II
Expected Lifetime	˜20-50 years	˜20 minutes - 1 day
Formation Speed	˜8 weeks	<1 minute
Formation Frequency	˜years	˜minutes
Turnover forecast	˜millions €s	<100 €

Tab. 2.2: VO classification criteria

and automotive industries as well as the high-tech industry with several years lifetime at its lower end. A detailed example follows later in this section acting on the business opportunity of a government issued tender for a plane upgrade contract. A systems integrator, acting as the VO manager, forms a VO with specialised part/subcomponent suppliers, designers and other service providers.

- **Class II** - This class encompasses short-lived, dynamic VOs cycling through all phases in mere minutes. Typical examples are set in the Ad-Hoc service provisioning (AH) business domain, for instance offering roaming services over WLAN to tourists. VO members are mainly telecommunication, (mobile) operator and service organisations. Usually, the local operator acts as the VO manager, forming the VO with the customer's home operator and the providers of the requested services, for instance tourist information or weather forecast services. Service request-response cycles in the VO's operation phase are brief, taking only seconds. The remainder of the VO lifetime is dedicated to billing during the dissolution phase.

Classes I and II are of most interest for this work due to their extreme properties. Many other VO types can be classified in between I and II, though boundaries are never sharply differentiated and none of the analysed types showed radically new properties. VOs from the eLearning domain, denoted as Class III in

Figure 2.4, with a learning centre and several specialised learning content providers are more related to Class I, but longer-lived. Credit Unions and VOs formed by emergency service organisations to react upon a natural catastrophe or other severe emergency are right in between both classes.

The next subsection now describes a plane upgrade contract application scenario of a CE VO followed by an AH VO centering around a tourist who invokes and receives services through a VO's network. In the following chapters, these scenarios will be referred to as the aerospace and telco scenarios. Each of the scenarios is segmented into the five life cycle phases that are individually analysed in particular with respect to trust requirements.

Aerospace Application Scenario

The VO's business opportunity emerges when a larger airline, the VO's customer, publishes a tender for a plane upgrade contract. This undertaking entails plane design upgrades, their validation followed by manufacture and replacement of parts and subcomponents of a large subset of an airline's fleet. One organisation can not address this opportunity alone which is why an aerospace system's integrator such as British Aerospace Systems (BAE) who intends to submit a quote to this tender needs to assemble a consortium of VO members. The system's integrator in this example enacts the VOM role. A set of VO members such as design specialists and design simulation experts, storage providers for the large amounts of updated design data as well as subcomponent manufacturers, part suppliers and carriers are needed to complement the VO consortium. In this example, the required business roles are shown in Figure 2.5. Business roles denote the (potential) VO member's business expertise in contrast to the two VO roles, VO Manager (VOM) and VO Member (M).

The Airline owns and maintains a fleet of aircraft. In the role of the customer it issues a tender for an aircraft upgrade. A system integrator (Production Consultant) intending to submit a quote to this tender knows from his own expertise which further roles have to be filled with specialised VO Members. The VOM's duty is to conduct the VO's resource planning, its management, and also to maintain and expose the product design database in a controlled fashion to the VO. Upgrading a plane involves a design upgrade as well, therefore a design specialist role is needed. In a SOA, this specialist also exposes a service, the analysis provider service. New and upgraded designs require a validation that is done in simulation runs. Due to the large amount of computing power needed to run simulations e.g. on a wing design upgrade, these are not performed by the design specialist role but by a dedicated High Performance Computing (HPC) Provider. New designs can be uploaded to and simulation results retrieved from an exposed HPC service. Airplane designs tend to require a large volume of storage space. To avoid unnecessary transfers and ascertain reliable storage, a specialised storage provider role tends to this data. Integrator and design specialist directly store design data through a storage service within the storage provider's domain. In turn, the HPC provider scheduled to run simulations retrieves the new designs directly through the storage service.

In summary, forming the sample aerospace scenario to answer an airline's tender the roles of a system integrator (the VOM), a design specialist, a HPC and a storage provider are needed.

The following paragraphs guide through the five VO life cycle phases, as depicted in Figure 2.2.

During the *identification phase*, a systems integrator starts forming the VO consortium in response to the airline's tender. First potential VO Members able to enact the required business roles are identified. In Class I VOs, a subset of potential partners may already be known from previous interactions due to the VO's longevity or supporting services such as directories are queried for potential partners fitting a specific VO role.

During the *formation phase*, trust relationships among the future VO members are formalised. There-

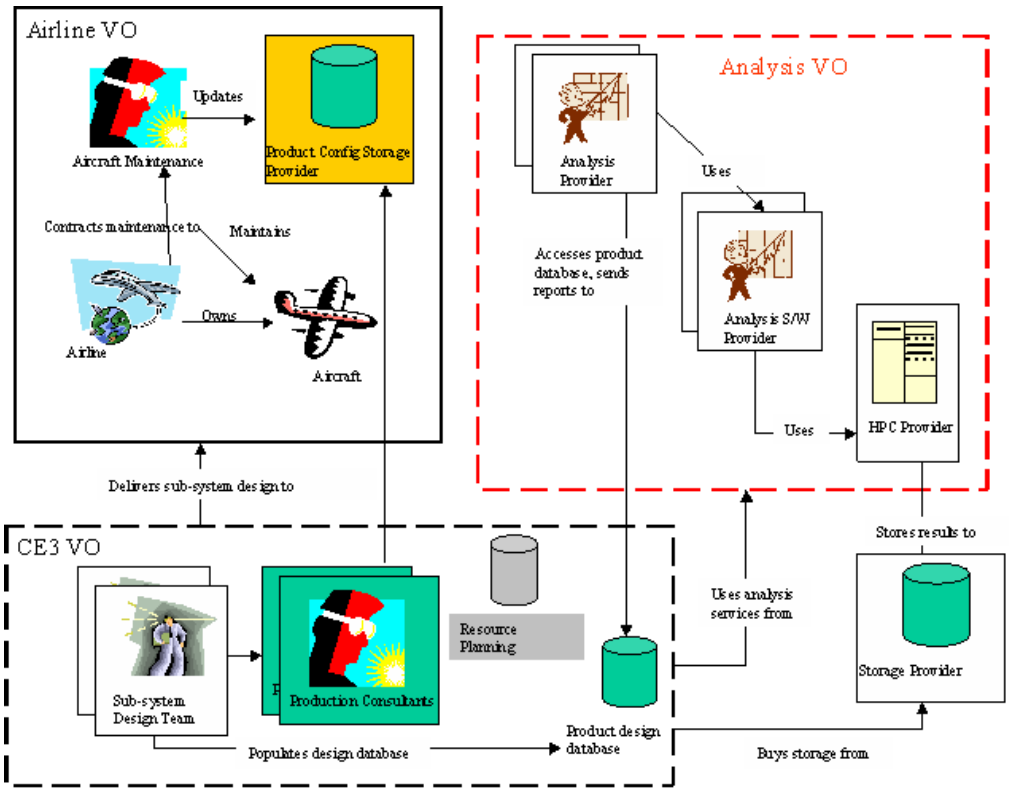

Fig. 2.5: Sample Collaborative Engineering VO (Source: TrustCoM Project/BAE Systems)

fore, trust management has the biggest impact in this phase. The systems integrator has to select a subset of the most trustworthy VO Members from a set of potential VO members for each business role. Typically, many candidates are available for each business role. Traditionally, in long lived, recurring and static business relationships, trust relationships last longer than one business transaction. Transactions are conducted rather bi-laterally than multi-laterally. A buyer selected the supplier he knew was trustworthy in the past. In VO environments, organisations tend to have a faster lifecycle, new players arrive in a market in higher rates and mergers lead to changes in organisational structures. In consequence, an organisation's trustworthiness changes with the development, must frequently be re-assessed and one can not solely rely on own experiences with a limited set of business partners. A reputation system for VO environments should be able to provide trust based decision support to quickly select the most trustworthy VO member from a set of potentials. Trustworthy hereby means, according to the adopted trust definition for this thesis, the potential VO member who will most likely perform reliably as expected by the VO's business role definition during the VO operational phase. The reputation system should constantly observe trust sources for each potential VO member and each role, design specialist, HPC and storage provider. Since this will lead to a large dataset from multiple trust sources, these data needs to be aggregated to a level that supports the VOM in his selection. The VOM can then compare for each business role each potential VO member's reputation that was obtained from the reputation system. Organisations with the highest reputation are then selected. In aerospace scenarios, this selection process is mainly driven by environmental and financial considerations. Environmental hereby means that it is important for the VOM if a storage provider or HPC puts his datacenter in an naturally stable region without earthquakes and if part manufacturers and carriers place their manufacturing plants and logistics centres in a region with high quality infrastructure. Financial considerations either mean the already stated fact that many supplier choices are driven by daily price schemes, but it also means that a financially stable VO Member will be able to perform his duty throughout the years long aerospace VO's lifetime. In the latter case, a VOM is not required to conduct the costly undertaking of a member replacement. During the remainder of the Formation Phase other preparatory VO management actions are taken. The selected VO members first have to be informed about the choice and must agree to join the VO [1]. Contracts and Service Level Agreements (SLAs) are negotiated and agreed upon. Furthermore, all VO members agree to a common business choreography [121] specifying the work each member will perform during the operation phase. A high demand on the ICT infrastructures of all collaborating organisations is imposed, especially on the security infrastructures. In general, each of the sovereign organisations has its own traditional or hard security infrastructure in place for e.g. policy and identity management, authorisation, etc. Those are typically not compatible with each other and require a high integration effort. A time intensive and financially expensive effort which has to be taken since the organisations need to collaborate seamlessly during the operation phase by exposing resources such as service endpoints and jointly enact distributed Business Processes (BPs). By only selecting the most trustworthy organisations as VO members, the risk of performing a costly ICT integration with a later expelled misperforming VO member in vain is greatly reduced. Further timely trust management support during the following VO phases can limit this risk even more, also in the case of inevitable VO Member replacement.

With the *formation phase*'s successful completion, the VO can commence its work upgrading the plane design and enters the operation phase. According to the agreed upon choreography, the VO Members commence enacting their BPs. Since trust is a subjective, dynamic property, further trust requirements emerge in this phase. A VO aiming at upgrading an airplane may last for several years. Most of this time is devoted to the operation phase. During this time, the reputation - and therefore trustworthiness - of VO members may undergo radical changes. The systems integrator orders a design upgrade from the design specialist. When the milestone "design" is ready after several months, the design specialist is due to submit the important

deliverable "design data" to the storage service. Questioning the storage provider's previously high reputation, the design specialist may query [7] the reputation system for a sudden decrease in the storage provider's trustworthiness. Having constantly observed the storage provider's trust sources, for instance an increased frequency in data losses would have been detected and a negative impact on the reputation value the consequence.

In case, a severe decrease of a VO member's reputation value or other case of misperformance were detected the VO can enter the *evolution phase* and change its structure. Such changes may require to perform some of the previous tasks again such as negotiating new contracts in case of a partner replacement. It is important to note that while replacing a partner is again a costly undertaking involving time delay and optionally penalties, the risk to continue with an unreliable VO member in a key business role may be even greater. Continuing the example, if the design specialist submits the design data without further scrutiny to the now untrustworthy storage provider validating simulation runs will commence, each result will be returned to the storage provider. An accidental data loss would throw the VO's plan back for several months, contract breaches will require penalties to be paid and eventually, the storage provider may have to be replaced anyway. Since Trust is a subjective property, two VO members in general do not trust another to the same extent. VOs act globally, a US based design specialist may have high trust in an Indian storage provider while a German design specialist needs to consider data protection laws that are not met in India. Therefore, the Germany based organisation has lower trust in the Indian storage provider than the US based one.

Successful or unsuccessful, each VO terminates at some point in time. Before that, it reaches the *dissolution phase*. Goods, data and during the VO's Operation Phase manufactured products are dispersed according to the terms in the contractual agreements. Finally billing and payments may take place. For trust management, its evaluation becomes important. The reputation system may have been invoked several times by different VO members, returning responses to each query. While the VO members will disband and enter other business relationships, retaining or abandoning established trust relationships, the reputation system will prevail and perform its duty for other VOs. To improve future queries for the same organisations as trustees, feedback about previous response throughout the VO's lifetime is a valuable input. Feedback may be actively gathered by the reputation system or its owner after a predefined time period after the query or at the end of the VO.

Telco Application Scenario

The telco application scenario is a more realistic version [111, 32] of the futuristic ISTAG ambient intelligence application scenario "Mariah, the Roadwarrior" from their vision for 2010 [31]. Figure 2.6 illustrates the setting and business roles.

In this VO, a business traveller acts as the VO's customer. He travels in a foreign country and carries a mobile device with wireless communication capabilities with him. Such devices range from mobile phones over Personal Digital Assistants (PDAs) to laptop computers that may communicate over wide area bearers (GSM, UMTS) or locally relying on Wireless LAN (WLAN). In case of a WLAN supported communication, a local gastronomy establishment, a cafe or restaurant, playing the business role of an access provider may deliver the needed network access by providing a public access point. The business traveller accesses the network with the intent to consume a variety of travel oriented services providing tourist information about local events and weather as well as his regularly accessed services offering access to his mail inbox

[7] Trust represents a bidirectional relationship. While it is not in general symmetric, trustor and trustee may change. In theory, any organisation may query the reputation system given the trustee's trust sources stated in query are observed

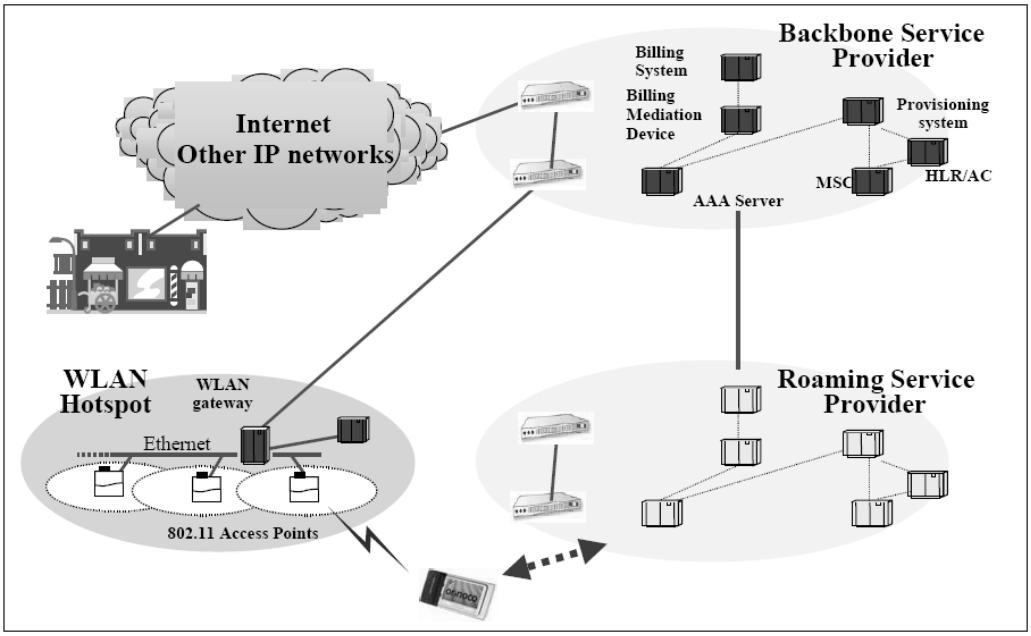

Fig. 2.6: Sample Ad-Hoc Service Provisioning VO (Source: TrustCoM Project/BT)

or displaying the favourite stock charts. Some service providers may charge for their services, others offer their services for free. To address the business opportunity of providing services to the business traveller, a dynamic VO is quickly formed. The restaurant, the traveller resides in, may or may not be part of the VO, mainly depending on the offered network infrastructure, if it is charged for. The restaurant has a wide area connection available over its WLAN gateway. The backbone service provider, playing the business role of a local operator, managing this connection becomes part of the VO as does the traveller's home operator providing regularly accessed services. The latter usually relies on an Internet connection to the local operator. Other organisations playing the business role of content service providers may also be required to join the VO if the traveller invokes their specialised services. It is assumed that these services are not individually discovered and invoked by the customer, these are rather presented in aggregated form, e.g. embedded within a portal presentation layer such as a personal portal page, by the home operator. In case of the traveller using e.g. a UMTS connection to the local operator, the restaurant's communication infrastructure is not used, but a Roaming Service Provider playing the business role of an access provider is required to join the VO. Since it would be rather bothering for a business traveller to pay each small amount for a consumed service and the used bandwidth from different operator networks separately, one of the operator's billing system, in most cases the home operator, is used to provide one consolidated bill to the traveller for one network access. This operator also enacts the role of the VOM. In summary, when forming the sample telco scenario, answering the business traveller's need to access services, the business roles of a local and a home operator are required. Depending on the type of network access, an access provider may optionally join the VO.

The following paragraphs again guide through the five VO life cycle phases.

During the *identification phase*, the potential VO Members are identified. The organisations playing the

role of access provider and home operator are already chosen by the customer, therefore no selection process is required in this case. The home operator maintains a list of local operators per region, sometimes already with pre-established umbrella contracts but still variable price schemes that may vary even from hour to hour. Depending on the customer's service needs, a candidate set for required content service provides is compiled. Many electronic services, especially offering travel, event, media or other entertainment content, are offered by a large set of service providers in the same manner.

With having potential VO Members for each business role available, the *formation phase* commences. On the infrastructure level, the home operator selects the local operator, a choice mainly driven by financial and operational considerations. The operator with the lowest price for that time is chosen having also the best operational qualities which addresses his network infrastructure allowing for uninterrupted and timely communication at sufficient bandwidth. Since the VO in the telco domain is about satisfying the customer's service needs, the selection of content providers is more crucial for the VO's success. Content providers may now be selected by the customer, the business traveller himself, or based on his preferences by his home operator. The former imposes considerable overhead on the customer who may only be interested in a quick glance at local weather or events which is why the following description focuses on the latter. The home operator performing the content service provider selection on behalf of the customer based on his preferences is able to achieve a higher degree of automation and mirrors current best practice. A Yahoo or MSN customer simply sees e.g. a weather chart in his portal page, but does not know which service provider delivers the content. The home operator's selection is mainly based on operational criteria since complete, timely and high quality delivery of service content is of the utmost importance. Considering replacement of a misperforming VO Member during a later VO phase is not of much concern in this VO class due to its brief lifetime and the inherent dynamic setup, that e.g. would allow for a cheap and seamless content service provider replacement. For that reason, the *evolution phase* is omitted in this scenario.

Having selected the VO Members, the *operational phase* now commences. This phase in the AH scenario is typically very brief and entails the delivery of the requested service content to the business traveller. This may happen e.g. be rendering his personal portal page in his device's web browser. The traveller may further on drill down into some service offerings, for instance book cinema tickets or prolongue his hotel reservation. When he drops the network connection, the operational phase is finished.

The *dissolution phase* in the telco scenario is quite extended compared to the operational phase's duration and can last several time longer. This contrasts to the aerospace VO's ratio, were the dissolution phase is much briefer than the operational phase. In the telco scenario, mainly the billing and the dispersal of the payment among the VO Members takes place. Since the home operator typically receives the full payment, it needs to be split as agreed upon in the contract during the formation phase between the VO Members.

2.3.2 Conclusion from the Application Scenario Analysis

Trust and reputation management are viable candidates to compensate for the loss of speed during the VO formation phase due to the difficult alignment of ICT, especially traditional hard security, infrastructures. Trust and reputation management hereby complement traditional hard security measures by providing soft security measures before hard security is established. Trust is a concept which is difficult to define in an IT environment, since such a multi-facetted concept is difficult to capture in technical terms a machine can work with. Due to the ICT dependency of VOs, this problem has to be tackled. Nevertheless, trust has a huge potential to capture an extensive relationship part of business interactions.

For the VOM, finding the right VO member organisation is a problem which can not be solved by hard security measures such as policy and access control infrastructures alone. It will not be cost effective nor beneficial for the overall security of the entire VO ecosystem to radically adapt "hard" security measures, e.g.

a member's internal access control and policy infrastructures for the temporary VO membership. Therefore, trust in a potential business partner is an important factor. It has to be noted that trust in a VO context does not mean "blind trust" in a sense that one VO member would uncontrollably expose his assets to another VO member of a high reputation. Based on a reputation evaluation, e.g. by obtaining a potential VO member's reputation value from a reputation system, a sensible organisation would employ for instance risk management best practices, such as from Operation Risk Management (ORM) [37] to mitigate any identified risks before entering a VO relationship especially with a less trusted partner. The decision making process if a potential VO member is chosen or not centres around weighting trust and risk factors against economical factors such as turnover forecasts or expected profit margins earned by jointly addressing the VO's business opportunity. The approach to provide automated reputation based decision support to a VOM meets the VO's formation speed requirements. It is also a more tolerating approach fostering VO relationships since the possibility that potential fraudulent and misbehaving partners might be selected is not denied, the VO will adapt to limit the damage such a partner can do for the benefit of the collaboration. [92] refers to this fact as *There shall never be a key that uncritically opens up all locks on the system.* arguing hereby for "soft", social security. In a sense, "soft" security in this approach helps to bootstrap "hard" security which would mean the integration of security infrastructures.

For the partner selection addressing for instance a government military tender, legal requirements may demand the resource intensive adaptation of the partner's "hard" security infrastructures. Not so in the VO scenario at hand, but there will still be remaining risks such as delivery delays of a key supplier in the VO. Those can be addressed with an accordingly formulated contract, stating for instance penalty clauses mitigating the risks.

2.4 Summary

This chapter analysed and discussed related work from the trust and reputation management research fields. Results of this section are directly influencing the design of the STORE reputation system in the following Chapters 3 and 4. The provided mathematical background in Bayesian theory and Bayes Networks as well as their practical use foster the understanding of the trust model that is underlying the STORE design. STORE's evaluation in Chapter 6 benefits from the analysis of related work in agent based simulation, especially cases where this methodology was previously applied to evaluate reputation systems.
Before a system such as STORE is designed, its application domain and problem definition must be set and analysed. Section 2.3.1 served exactly this purpose, introducing VOs as STORE's application domain and defining the problem domain with two specific scenarios from collaborative engineering and ad-hoc service provisioning VO domains. These scenarios are chosen because of their properties putting them at opposite ends of the VO classification. They are revisited in Chapter 6 and deliver the setting for STORE's evaluation.

3. THE STORE REPUTATION SYSTEM - DESIGN TIME

Chapter 3 and 4 introduce the technical core contribution of this work. It consists of a design and runtime part that, put together, form the model and architecture of the STORE reputation system. Distinguishing between the abstract model and the instantiated architecture is the reason for spreading the technical contribution over two chapters.

The abstract model consists of static design artefacts, such as a Unified Modeling language (UML) class diagram capturing the TI taxonomy, the TI model along with indicator specific attributes and the BN based aggregation methodology. For the latter, certain model properties such as the BN topology are also runtime independent.

Whenever a reputation system based on the STORE design is employed in an application scenario such as an aerospace or telco one, certain measures, e.g. configuration or TI selection, have to be taken into account at designtime, prior to the system's "going live" event which defines the boundary between the system's design and runtime lifecycle phase. This chapter introduces and describes the STORE components and artefacts of primary relevance at designtime which are

- The Taxonomy of TIs
- The TI aggregation methodology and their attributes

The taxonomy and the TI model form the trust model underlying STORE and are this thesis' core contributions in this chapter.

The STORE reputation system is intended to provide automated trust based decision support to a trustor, e.g. a VOM, for selecting a subset of trustees, e.g. (potential) VO members, from a larger set of potential business partners. To achieve this goal, STORE needs to provide a trust model that is able to capture each trustee's behaviour including its dynamic changes over time, in particular a measure for their expected reliability. The STORE trust model described in this chapter follows a novel stochastic approach. Each TI capturing one organisation's property inherently characterising one of its trust aspects is modelled using a probability density function. TIs are aggregated towards a reputation measure following a stochastic approach that is based on a Bayes Network (BN).

STORE's architecture and its instantiation at runtime is detailed in the following chapter 4.

3.1 Taxonomy of Trust Indicators

In [75], Luhmann said

> "Trust, on the other hand, requires a previous engagement on your part. It presupposes a situation of risk.

Taking this approach which essentially means that trust ”drives” risk fits well into a VO environment dealing with collaborating organisations. Without having had a previous engagement with a potential VO member, a VOM does not know if a selected member will turn out as a bad decision or not. That is the risk, a VOM must take. He can avoid this risk only by neglecting all the advantages of forming a VO which are the expected revenues from the business opportunity. Providing an automated trust based decision support with a reputation system compensates for missing previous engagements and helps to drive the risk of selecting an untrustworthy partner down. The STORE reputation system is designed as a central system. Since its reputation service is intended to be used by a wider range of trustors, not only one, it is important to discuss the ownership of such a system since this provides the answer to the question ”Why trust the STORE reputation system and why trust Trust Indicators?” . If the system belongs to an involved party or role, trustor or trustee, the accuracy and truthfulness of the delivered reputation measure becomes questionable. In short, the trustworthiness of the STORE reputation system itself decreases. Such ownership situations give rise to attacks and manipulations of the system itself as discussed in Chapter 5. A central reputation system is best owned by an explicitly trusted, independent third party, a Trusted Third Party (TTP). This is assumed for the following chapters. This assumption is later weakened when discussing the STORE threat model and stated otherwise in these cases.

Inspired by the close relationship between trust and risk, a set of *Trust Indicator*s (TIs) is defined with the following properties:

Definition 4: A Trust Indicator (TI)

- indicates a trustee’s trustworthy behaviour in a specific trust aspect, e.g. a certain area of the trustee’s organisation.
- can be observed and delivers measured data in regular intervals, accounting for a trustee’s behavioural changes over time
- can exhibit a stochastic behaviour, the interesting type for this work, but can also behave deterministically
- is the basis for a trustee’s predicted behavioural development for the future

Taking a top-down approach, the following subsections classify TIs according to their trust aspects leading to an extensible TI taxonomy. The classification effort commences by identifying the five abstract TI top-level classes. Drilling down into each top-level-class, subclasses are identified. The classification results are finally summarised as an UML model. Within the subclasses, the detailed model of concrete, individual TIs with all their attributes and an example set of TIs with relevance for the following chapters as well conclude this chapter. The entire TI list, as yet defined, is provided in the Appendix A.2. This section concludes with extensibility aspects of the TI taxonomy. Neither the set of TI classes, nor the list of individual TIs is supposed to be final and can in fact be extended.

3.1.1 Taxonomy

The classification effort commences by identifying the top-level Trust Classes (TCs), grouping individual TIs that stem from or characterise the same trust aspect of an organisation. The end result is depicted in Figure 3.1.

VOs have the fact in common that collaborating member organisations address a common business goal. While doing so, operational aspects arise for instance when goods or information are exchanged among

member organisations. In an aerospace scenario, the design analyst requires data from the storage provider. Later, part manufacturers e.g. assembling a plane's wing require shipments from specialised component manufacturers. Telco scenarios are mostly information driven, the customer retrieves the requested service content or operators exchange billing data. Operational aspects of trust entail notions such as a timely delivery, availability of systems and services, quick reaction to product complaints and other quality aspects. They are modelled as *operational TIs*.

TI Class 1: Operational TIs

Since trust is inherently related to risk, existing operational risk categories heavily influence the further subclassification of this trust class. On an abstract level, operational risk is commonly divided to be derived from *staff, technology, process* and *environment* [70]. Technology and process refers to risks surging in the operational processes of a firm. Starting from there, the following subclassification of operational TIs can be taken:

- **Procurement TIs** - model trustworthy behaviour in sales order, order to cash and other procurement related processes.
- **Inventory TIs** - characterise trustworthiness in order to stock, supply chain planning and similar stock and product inventory processes.
- **Production TIs** - describe trustworthy behaviour on the production side of supply chain planning as well as properties such as product quality and production rates.
- **Shipping TIs** - model trustworthiness in shipping processes, if shipments are sent in time, arrive without delay and with the right quantity.
- **Systems TIs** - consist of indicators modelling technical quality characteristics of especially externally available systems for participation in collaborative processes such as system availability and failure rates.
- **Legal TIs** - model in contrast to the external regulations TIs to which extent an organisation internally takes legal measures and controls for instance to achieve IP protection or legal compliance[33].
- **Service TIs** - address an organisation's service offering in general, not only focusing on technical services. This subclass encompasses properties such as service response times but also indirect quality measurements like service complaint rates from collaborating organisations.
- **Sales TIs** - model trustworthy behaviour in sales processes. Besides volume and throughput of a sales pipeline, properties such as the complaint rate to individual sales incidents are of relevance.

Operational TIs are of importance for all types of VO scenarios simply due to the fact that each VO's operational phase, to some extent, has to deal with distribution of goods or information. This fact and the wealth of work in related fields lead to the most extensive subclassification throughout the entire TI taxonomy. Most of the operational TIs characterise the day to day business reliability. These TIs can therefore be observed frequently and frequently deliver fresh data on a daily to weekly basis.

TI Class 2: Organisational TIs

A further subclassification according to the functional units of the firm may seem applicable, but cannot generally be provided, since a multitude of organisations and hence distinct organizational structures in different VO application domains are under consideration. Staff points to more hidden risks caused by human behaviour. These can occur on different decision levels, strategic, managerial or simply personnel level, having an impact on an organisation's trustworthiness.

These properties are denoted as the trust class of *organisational TIs* with the following subclassification:

- **Strategy TIs** model the organisation's trustworthiness for the longer term by measuring e.g. the failure rate of strategic decisions or the rate of abolished strategies.
- **Management TIs** model the organisation's management trustworthiness by measuring the management to employee ratio, management growth rates or applied management methodologies.
- **Personnel TIs** characterise the organisation's trustworthiness on the employee level. Employee turnover, the rate know-how bearers leave the company and similar TIs comprise this subclass.
- **Innovation TIs** model the creativity that is inherent to an organisation's personnel by measuring for instance the rate of successful patent applications, frequency of announced major product versions and innovations.

Organisational TIs are especially relevant for longer lasting VOs. This class of TIs can only be observed and delivers data on a mid to long term basis. More employee related TIs are estimated to deliver fresh data observations on a weekly to monthly basis while strategy TIs can be observed in a monthly or longer time period.

TI Class 3: External TIs

The trust class of *external TIs* refers to influences on an organisation's trustworthy behaviour and reliability from the outside. These can be other parties outside the VO like customers or competitors, regional or global legislation but also non-entities like the general economic environment, labour and factor markets or the natural resources and dangers.

- **Competition TIs** model the fierceness of a competitive environment, e.g. a VO application domain, by measuring the rate of emerging organisations and bankruptcies as well as the relative organisation's competitiveness as determined for instance by peer ratings.
- **Economy TIs** model the regional economic health based on economy and market indices.
- **Regulations TI** model to which degree an organisation is legally compliant to international regulations (Sarbanes-Oxley, Basel III for instance), e.g. to which degree security controls to avoid segregation of duty, are in place. This subclass poses a challenge at the technical possibilities of defining and modelling such TIs in practice.
- **Environment TIs** entail aspects such as the impact of infrastructure, natural resources and stability on an organisation's trustworthiness. These aspects are for instance captured by national indices or ratings.

External TIs model more long term influences on an organisation's trustworthiness. These TIs are expected to deliver fresh data observations on a quarterly to yearly basis. They typically consist of data sources delivering already highly aggregated data measurements, for instance a rating capturing an entire nation's stability with respect to natural disasters.

TI Class 4: Financial TIs

Furthermore, the trust class of *financial TIs* captures an organisation's reliability and trustworthiness which becomes most apparent in the worst case if a firm fails due to financial bankruptcy. Financial information can be based on balance sheet data or from indirect, pre-aggregated measures like stock market figures. Popular indicators for financial performance measuring are for example the cash flow quote, economic value added (EVA) or earnings per share ratio (EPS) [106].

- **Balance TIs** model the subclass of financial TIs that are private to an organisation.
- **Stock TIs** model in contrast to Balance TIs an organisation's trustworthiness based on publicly available financial data.

Financial TIs are usually observed and updated in weekly to monthly periods. While Stock TIs are publicly available, they can be safely disclosed since they consist of aggregated financial information, it is much harder to obtain data for Balance TIs. The latter contribute very accurately to an organisation's reputation measure since this trust subclass consists of individual, detailed TIs that allow to infer and predict an organisation's tactics or even strategy. These TIs are therefore kept private and are difficult to observe for a principal outside the assessed organisation. Section 4.3 follows up on the challenge of observing and obtaining confidential TI data for a reputation measure.

TI Class 5: Third Party TIs

At last, reliability information may also stem from a third party. Various commercial information providers, such as rating agencies, have tackled the task of giving widespread and meaningful ratings about potential business partners. Prominent examples are financial stock ratings from Standard & Poors[1], Moodys[2] as well as the company database maintained by Dun&Bradstreet[3] amongst others providing extensive information about credit-worthiness of organisations. These rating activities are based on obtained data for mostly financial indicators similar to or the same as the presented TIs. This also includes confidential indicators that are related e.g. to Balance TIs which motivates their inclusion into the TI taxonomy despite the difficulty of their external observation. The mentioned information providers deliver their rating information in different forms. In each case, the information provider acts as a Trusted Third Party (TTP) that is explicitly trusted by all involved parties. A customer, the trustor, may be interested in a specific potential business partner, the trustee, and asks for a detailed report. The answer to such a request consists of a larger document and an expensive fee is charged. Due to the information provider's status as a TTP, the trustee discloses private, e.g. financial or organisational, information which, in the report, contributes to aggregated data and is not disclosed as such. Another delivery form is via an exposed, technical service interface. A customer integrates his ICT infrastructure with the TTP's web service that delivers a highly aggregated rating value about a trustee to the customer. A third form is the delivery of peer ratings to a group of subscribed customers.

[1] http://www.standardandpoors.com/
[2] http://www.moodys.com/
[3] http://www.dnb.com/

Each participating customer signs an agreement that he is willing to disclose observations of the same indicators, even confidential ones. The data is obtained by the TTP, anonymised and aggregated into a so-called peer rating report. Each subscribed customer receives in regular intervals a customer specific report from the information provider, comparing himself to a peer group, e.g. the set of all other subscribed customers from the same application domain. Data from the peer group is only shown in highly aggregated form. Input from TTP information providers as well as reputation measures from other instances of the STORE reputation system, can be integrated as a *Third Party (TP) TI* carrying aggregated reputation information content [112]. Such information may consist of already pre-aggregated reputation values from other reputation systems or recommendation values from domain-specific recommender systems. This gives rise to the definition of the third trust subclass of *TP TIs*:

- **TP System TIs** model input from external information providers.
- **TP Instance TIs** model input of other STORE reputation system instances that may be specialised on a particular application or VO domain.
- **TP Certificates TIs** model external influences on an organisation's trustworthiness stemming from certification activities such as ISO and SAS certificates.

TP TIs are expected to follow their individual update intervals that are determined by an information provider's report frequency and the charge demanded for his information. More comprehensive reports are typically published on a monthly to quarterly basis while individual ratings are updated more frequently.

Figure 3.1 depicts the results of classification exercise as the TI Taxonomy in form of a UML class diagram. The subcategories, that are specialisations of the top-level trust classes, developed following to a bottom-up approach by first collecting a number of TI candidates. These candidates were then analysed in detail for relevance in describing an organisation's trustworthiness and therefore their ability to contribute to a reputation measure. In most cases, this analysis could be based on relevant literature from the previously stated research fields closely related to trust and reputation management. Risk theory and risk management literature such as [106, 9, 60] confirmed most TI candidates as well as other specialised areas such as financial risk management [47, 7], operational risk management [70, 24, 34] and there particularly supply chain risk management [71, 18, 20]. The resulting set of, so far, 146 individual TIs was then clustered into trust classes and subclasses. The set of the five trust classes could already be conceived from trust and reputation literature due to the consensus of coarse grained aspects that influence trustworthy behaviour. Their setup did not change during the TI clustering effort. In contrast to the trust classes, many subclasses changed or were added, especially in the class of Operational TIs. The end result, the set of 146 TIs and their classification is is available in Appendix A.2, Table A.1.

The taxonomy represents the reference framework for trust that serves as the basis for the STORE reputation system and its aggregation of a reputation measure.

The taxonomy encompasses the set of currently identified TIs contributing to a model of an organisation's trustworthy behaviour having survived the scrutiny of a thorough analysis. On the one hand, when implementing this model for a concrete application scenario such as an aerospace or telco scenario, not all TIs are required at once to be instantiated for a potential VO member's trust model. Section 4.2 in the following chapter deals with reputation system instance specific configuration and bootstrapping problems. On the other hand, the TI Taxonomy is intentionally designed to be extensible. If an application scenario specific trust facet is not captured by any of the existing TIs, a new one can be added to the taxonomy following the modelling approach introduced in the next section 3.1.2.

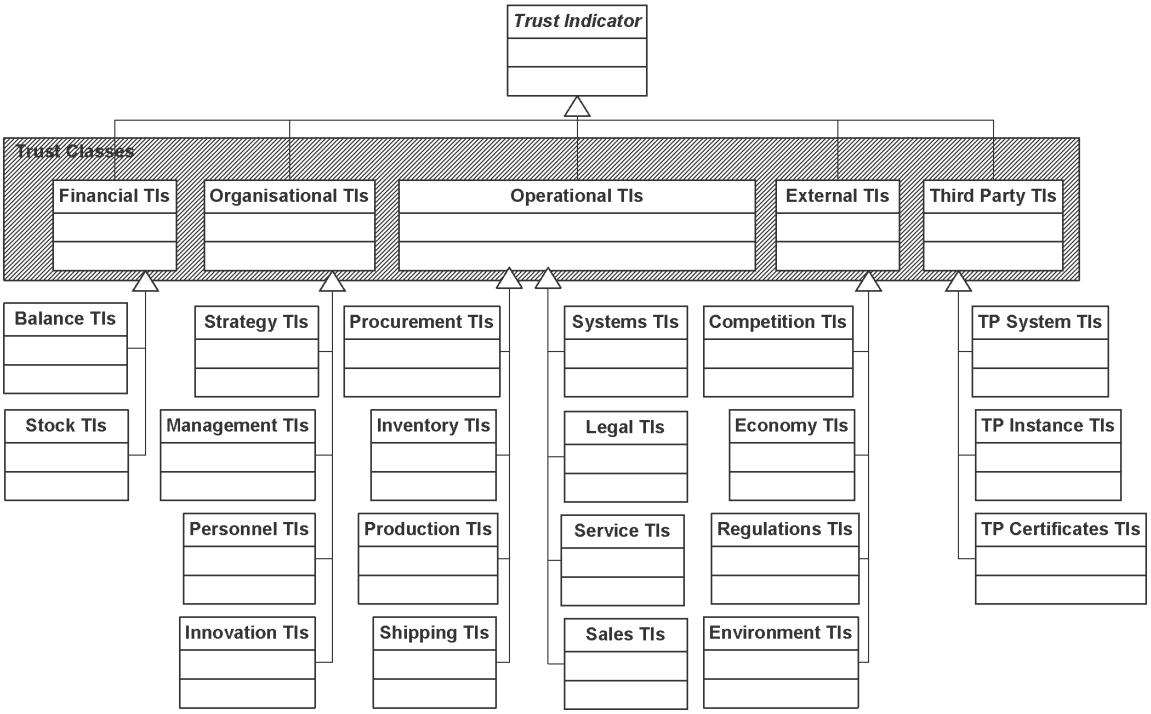

Fig. 3.1: TI Taxonomy

3.1.2 The TI Model

As stated in the beginning of this chapter, the STORE trust model follows a stochastic approach. Departing from the TI Taxonomy reference framework consisting of the set of all so far identified TIs, this section describes the model of an individual TI in more detail and focuses on TIs exhibiting a stochastic behaviour. Such TIs are modelled using probability density functions, taking a distribution assumption, that closely follows the TI's real behaviour. Such a stochastic model allows for better predictions of an observed TI's near and mid term development and therefore in turn, for an organisation's expected near and mid term trustworthy behaviour. Stochastically modelled TIs therefore capture the dynamic trust changes over time. While some of the TIs introduced in the following paragraphs clearly exhibit stochastic behaviour and fitting distribution assumptions could be derived from literature, others can not reasonably be modelled with density functions. These are considered as deterministic TI which still fit the model, e.g. by assuming an uninformed uniform distribution, and can be captured in STORE's TI model. But they contribute with a higher degree of uncertainty to a reputation measure and are therefore not in the focus of the following work.

Every TI is modelled according to a set of attributes that details its usage within the model and guides its implementation:

Name N. Every TI is uniquely identified by a name N.

Domain D. A TI is based on observations of a continuous or discrete variable x. The possible values of x

are the domain of the TI. A full set of observations $X = \{x_1, ..., x_n\}$, $x_i \in \Re$ have a maximum and minimum value, spanning the TI domain $D = [x_{min}, x_{max}]$.

States S. The TI measure is discretised by defining certain bounds $x_b \in D$ dividing the TI domain into s_{max} disjunct intervals, $S_j = [x_{b_{s-1}}, x_{b_j}]$, $s \in \{1, s_{max}\}$, so-called states S, in which observations x are obtained. The states S consolidate discrete and continuous measures towards a common discrete handling.

Update time period Δt_{upd}. TI observations is likely to arrive in TI specific observation periods. The attribute Δt_{upd} defines a fixed update period of how often STORE needs to update a TI.

Observation time period Δt_{obs}. The time period $\Delta t_{obs} = t_{obs_{end}} - t_{obs_{start}} = const$ defines the maximal time window to look into the past. Beyond that, observations are regarded to carry no more significance for current and predicted trustworthy behaviour. This attribute takes into account that a trustee's recent behaviour carries more significance for a reputation measure than past behaviour. Naturally, the observation time period must be greater than the update time period $\Delta t_{obs} \gg \Delta t_{upd}$.

Time weighting function ω. Among n observations x_i at times t_i , $i \in \{1, ..., n\}$ within the time window, old ones are less likely to carry information about future behavioural development than recent ones. Each TI incorporates a monotonically increasing weighting function $\omega(t) > 0$ with $\sum_{i=1}^{n} \omega(x_i, t_i) \stackrel{!}{=} n$, $\forall t \in [t_{obs_{start}}, t_{obs_{end}}]$, that implements forgetting of older observations and puts an emphasis on recent ones.

Empirical distribution E. Upon availability of fresh TI data, this newly observed data θ is assigned to states and counted to an empirical frequency distribution $E(\theta)$. $E(\theta)$ is primarily providing information about the latest observation about that particular TI's development.

Likelihood distribution L. Further on, every TI observation, due to its stochastic nature, follows a certain statistical distribution. L reflects the likelihood $L(X) = P(X|\theta)$, that X is the "real" parameter underlying the distribution of the TI, given a set of observations θ, [63]. Its distribution assumption itself has to be derived from statistical analysis or expert knowledge. The Likelihood distribution also takes the TI's historical data into account and can be calculated upon the availability of each new observation data set θ, taking the weighting function into account. To take forgetting into account, at first the time weighting function is applied to the Likelihood distribution's conditional probability $P(X|\theta)$, more concretely to the real parameter X, with $w = \omega(X, t)$.

Trust preference mapping π. In order to judge the level of trustworthiness displayed by a TI, an ordinal scale 1 to p_{max} is defined, where 1 represents the lowest and p_{max} the highest level of trust indicated by the TI. To compare heterogeneous TIs, each with its own domain and set of states, the scale remains the same for all TIs, $p_{max} = const.$ π defines a function $\pi \colon S \rightarrow \{1, ..., p_{max}\}$ mapping the TI's states S to the different levels of trust indicated by them. This mapping enables an expert to incorporate his knowledge on the particular TI domain.

To apply the STORE trust model to an organisation, a trustee, a subset of TIs from the taxonomy is selected at the system's designtime that are relevant to the intended application context. The context is determined e.g. by the VO application scenario, the trustee intends to participate in. The TI selection process can be conducted in several ways. A template based approach avoids putting a burden on trustor and trustee. The TTP owning the reputation system knows about TI modelling and its service requesters, e.g. the VOMs from specific VO application domains. The TTP, acting as a domain expert, knows about the

VOMs preferences which TIs are important in which class of VOs and can offer a preprepared template of TI. A VOM, naturally a business expert in its VO application domain, can also edit the template if required. A second possibility is a TI selection individually done for each trustee by a domain expert, either the VOM or the TTP owning the system. This approach requires a higher manual effort but may lead to higher quality reputation measures and therefore better decision support at runtime when a selected becomes instantiated as a TI instance. TIs have to meet the set of requirements stated at 1. The presented TI model meets all of these requirements. Selected TI instances model exactly one trustee, one organisation and contribute in a bidirectional trust relationship to a reputation measure provided by the STORE system to a trustor. Since each TI instance captures exactly one trustee's trust facet, the model takes the subjectivity of trust into account. The TIs themselves form the basis of the STORE trust model. Characterising a trustee's trustworthy behaviour they can be observed and deliver objective data in regular time periods. The TI Taxonomy and especially the TI attributes capture most information that is required for their implementation and instantiation. The remaining part belongs to the system's configuration, e.g. technical configurations from where TI data can be observed. The STORE system runtime itself building upon the presented model is highly automated. Due to the trust preference mapping, heterogeneous TIs become comparable and can, as described in Section 3.2 be aggregated towards a reputation measure a trustor can comprehend. Regularly updated TI observations capture changes in the trustee's trustworthy behaviour over time.

3.1.3 TI Updating

Having introduced the TI Taxonomy, TIs and their attributes were discussed. One of these attributes, the Likelihood Distribution, takes a distribution assumption for each TI. That means assuming a probabilistic model based on a density function how this TI will most likely develop in reality. On the other hand, the model receives input from the real world. Each TI instance corresponds to an organisation where it can be observed. The continuous process of updating a TI refers - on an abstract level - to the mechanism connecting these data observations to the probabilistic model. A naive approach could take the data observations as the only new input for the TI's assumed distribution. In that case, the distribution parameters would be unknown which is a regular problem in statistics. It can be dealt with by estimating the parameters. A well defined estimator of benign properties such that it is stable and unbiased, frequently a Maximum-Likelihood estimator can be used, delivers a parameter estimation of low variance. While being simple and straight forward, this approach has several disadvantages. The STORE model takes TI data from before the last observation update into account. This can be covered with the naive approach, but extensions taking past data into account make it more complex. TI observations sometimes suffer from defective sensors and systems at the observation's source or distorted communication lines. The result are incomplete or wrong data. A distribution parameter estimation is vulnerable against false data input and would lead to shifted parameters. Basing a reputation measure on such TIs would result in an ineffective decision support mechanism. For the TI update in the STORE reputation system an approach based on the Bayesian Update is used, see Equation (2.4) or [91] for more information. According to Bayesian theory, a prior distribution $P(\theta)$ represents an uncertainty distribution of the prior belief about the real value θ of the freshly observed TI data. In the case at hand, the prior can be derived from the empirical distribution $E(\theta)$ e.g. by calculating the relative frequency distribution $P(\theta) = \frac{1}{n}E(\theta)$. The posterior $P(\theta|X)$ represents the best knowledge of θ taking available historic data X of the observation interval's Δt_{obs} size into account. The posterior distribution is discretised over states S, and connected to the prior function with the likelihood $L(X) = P(X|\theta)$ via the Bayes Theorem, using 2.4 and 2.6:

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)} = \frac{P(X|\theta)P(\theta)}{\sum_{S} P(X|\theta)P(\theta)} \tag{3.1}$$

Following this approach, a realistic and probabilistically justified ”fit” between epistemic knowledge represented by the empirical distribution and incorporated assumptions represented by the likelihood function is achieved. As described in 3.1.2, the likelihood distribution captures with the distribution assumption the historic data. Fresh data observation are injected each time as a prior distribution of a higher uncertainty which will increase the prior's uncertainty. This occurs inevitably due to unsanitised data, entailing observation glitches, communication errors from obtaining the data and similar defects. The effect of these defects is mitigated while performing the Bayes update. They are combined with the Likelihood distribution's incorporated assumptions of previous observations and normalised by the denominator's marginalisation of θ bearing the uncertainty (2.6). Further aggregation towards a reputation measure is then conducted only on the basis of the posterior distribution.

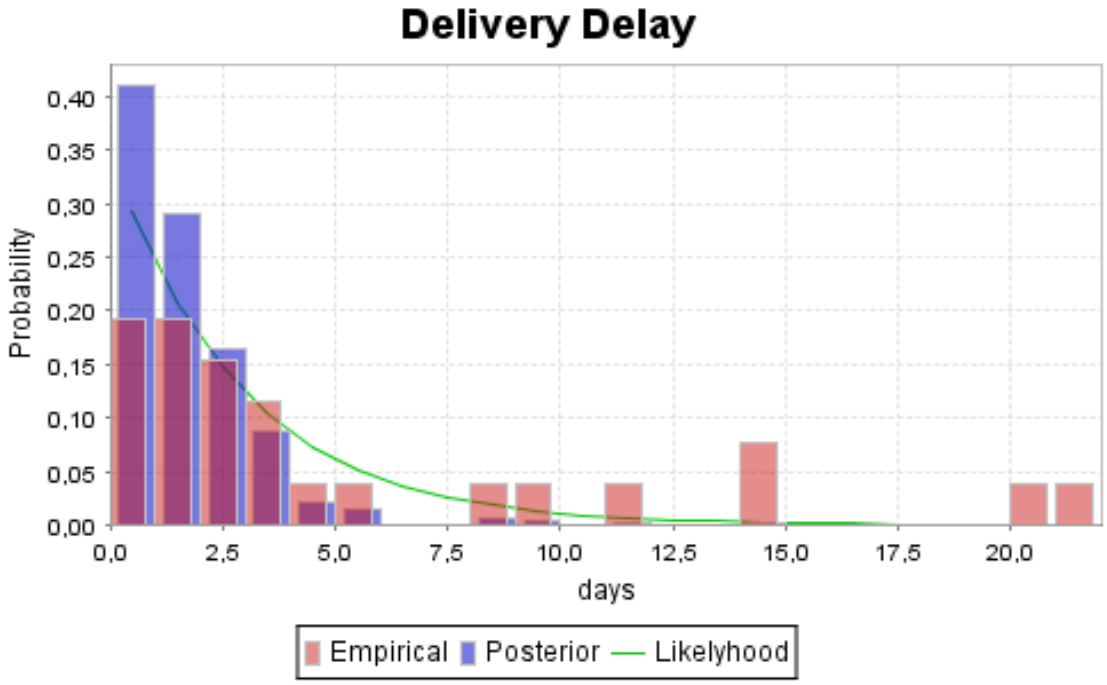

Fig. 3.2: Trust indicator updating

Figure 3.2 depicts the result of a Bayes update for the TI ”Delivery Delay” which is assumed to follow an Exponential distribution. Fresh observation entering the equation as the prior distribution show for instance an outlier around 14 days and two more beyond 20. The continuously drawn Likelihood distribution asserts this organisation a more trustworthy behaviour. According to the past performance, such long delays rarely or never occurred. The posterior distribution takes this into account and does not shift the distribution's

expectation value suddenly to higher delays because of isolated outliers, but shifts slowly. It has to be noted that the time weighting function puts an emphasis on more recent observations when computing the Likelihood distribution.

3.1.4 Examples for Trust Indicators

At this point, five examples for TIs are provided. These examples are intended to demonstrate the TI modelling using the TI Taxonomy's information during the STORE system's design time. These five TIs were selected for the following reasons:

- Coverage of all TI top-level trust classes.
- All exhibit a stochastic behaviour.
- Represent TIs modelling individual and (pre-)aggregated trust properties.
- Relevance for both aerospace and telco scenarios.

To reference a TI within the taxonomy from Figure 3.1, the following notation is applied for the following TI examples. First, the TI's trust class, then the subclass it belongs to is stated, delimited by a ":":

trust class:subclass.

For each example TI, the relevance for both VO scenarios is reasoned. The TIs were carefully chosen for their relevance in both VO scenarios, aerospace and telco, which is a requirement to ensure comparability of the evaluation across the scenarios - as is discussed in Chapter 6.

The *Cash Flow Margin TI* (CF) measures the Cash Flow from operating activities in relation to the organisation's net sales. The Cash Flow itself is a measure for the actual cash generated by a business. It can be observed, especially from Operational Risk Management, that a low Cash Flow Margin signifies a low default risk and therefore a trustworthy organisation [48]. The default risk originated in the microeconomic development of a stock company and is measured by a wide range of financial indicators. It is used to provide an aggregated overview of an organisation's mainly financial development to its stakeholders and how far legal constraints and regulations are met. CF is classified as *Financial:Balance*. It captures a rather individual financial trust aspect that is typically confidential. Adopting [53], CF is modelled using a lognormal distribution (A.1.4). A VO, by definition, addresses a business opportunity. Therefore, financial matters in general and their influence on an organisation's trustworthy behaviour are always relevant. CF in particular is a well suited indicator for an organisation's economic stability. Furthermore, CF (and other financial indicators) deliver a well suited input for forecasting a company's economic progression [48]. While being relevant for both VO scenarios, observing the CF TI is of more interest in a long lived aerospace scenarios. Changes in the CF's development are unlikely to happen within an AS VO's short lifetime and therefore only the momentary financial trustworthiness of a potential VO member can serve as decision support. This is also the case for the aerospace scenario, but due to its lifetime of several years, CF changes are likely to occur during the VO's lifetime. A VOM, periodically requesting updates of his VO member's reputation value, takes these changes into account for his decision making throughout the VO's operational phase. Changes to the worse may happen for instance due to a revealed fraud, to the better due to exceptionally good sales. The relevance of financial TIs in general increases with a VO's expected lifetime.

VOs and their participating organisations operate on a global scale across national and regional boundaries. Therefore they are subject to methodologies from macroeconomic risk management. [73] considers

the *Country Bond Default Spread TI* (CBS) as a reasonably aggregated measure capturing risks from the economic, political and social environment of an organisation's country. [26] shows furthermore how this measure encompasses a country's currency risks and interest rate volatility. It is calculated by aggregating the yields on bonds issued in a particular country. A higher CBS denotes a higher environmental risk and deems an organisation of that origin as less trustworthy. CBS, a highly aggregated, public TI [4], is classified as *External:Economy*. Empirical studies show, that fat-tailed distributions like the Student's t-distribution (A.1.4) or the Normal distribution (A.1.4) deliver a good distribution assumption for CBS [102, p. 58]. In general, VOs consists of geographically dispersed member organisations. This holds true for both VO scenarios. In an aerospace scenario, manual labour e.g. for assembling plane wing parts, is conducted in low cost countries while design activities take place in countries having specialised high-tech know how. The AS scenario includes at least the business travellers current location, a local and home operator and optionally content service providers residing in outsourcing countries. In the case of a global organisation, conducting business in more thane one country, two choices of modelling environmental trust aspects present themselves. One achieves better results if a specific organisation's country subsidiary participating in a particular VO can be identified. In that case, the CBS TI for only this country is modelled, introducing minimal variance and therefore uncertainty to the overall reputation aggregation. If the country can not be narrowed down, one CBS instance per participating country subsidiary can be modelled, where each CBS instance increases the overall uncertainty of the reputation measure. Again, the CBS TI carries more relevance for a long lived aerospace scenario. Since environmental risks captured by the CBS, for instance natural catastrophes or damages to a region's infrastructure, are unlikely to occur without a warning at all, a VO in the aerospace domain benefits more from continuous observation of this TI. In general, the CBS TI's relevance for a VO increases the higher a VO's members are geographically dispersed and the longer it is expected to last.

The *Complaint Rate TI* (CR) measures the amount of customer complaints (for a specific good or product) relative to the amount of (product or good) items sold. The CR reflects the quality of an offered service or shipped product. Organisations offering services or shipping goods possess a means of service management for post-sale customer complaint handling. Callcenters offering telephone hotline numbers or complaint addresses for postal or e-mail are examples for such means. The CR itself is subject of interpretation. [74] show how variations in number and type of complaints can be accredited to different customer groups. CR development is influenced by human behaviour, in particular the country of the customer's origin, customer group attributes (age, sex, etc.), the shipped product and other criteria. This interpretability of CR requires a dedicated trust preference mapping for each instance of this TI in STORE reputation instance modelling an organisation in a particular VO scenario. Since CR is only one of a set of aggregated TIs and changes of the system's calculated reputation measure inferred from changes in the CR are of more interest than its level, even a slightly off trust preference mapping would not render the STORE reputation system useless. CR is frequently perceived as an important qualitative component of an economic environmental measure [60] and therefore classified as *Operational:Quality*. The same authors also suggest the Erlang-k distribution (A.1.4) as a suitable distribution assumption. CR, as a quality measure for a service or product, reflects the generated utility throughout a VO collaboration. Since it can be equally applied to tangible products, such as wing parts in an aerospace scenario, and digital products, e.g. the answer from an invoked travel information or booking service, it is equally relevant for all VO scenarios. As a rate measure, it characterises an organisation and is independent of a VO's lifetime. Since the members of a telco scenario offer and consume digital services, the expected transaction volume per time interval is higher than in an aerospace scenario and the CR TI an important prediction factor for future trustworthy behaviour.

[4] for instance available on a per country basis from Onvista `http://www.onvista.de`

A well established quality measure for order or service fulfilment is the *Delivery Delay TI* (DD). [71] and [8] identify DD as one of the key indicators in performance management. DD measures the time delay between an expected, e.g. contractually agreed upon, delivery date of a service offering or shipped good and the actual delivery date. DD may have its causes in capacity overutilisation, production delays, distribution chain inefficiencies, communication delays and many more. It is a specific TI characterising the operational short term reliability, and therefore trustworthiness, of a trustee. DD is classified as *Operational:Shipping*. Already early literature suggested an Exponential distribution (A.1.4) assumption [87]. Following the same line of argument as with the CR TI, DD is equally relevant for both VO scenarios. It can be applied to delays in supply chains, e.g. when shipping tangible products such as plane components for central assembly, and as well to delayed service responses in a telco scenario, for instance when invoking a travel booking service or while the home operator aggregates the billing of other involved service providers.

With the increased reliance on IT systems in business relationships, availability of these systems becomes paramount. This is especially true for VO environments with their ever increasing ICT infrastructure integration where an unavailable system amounts for huge financial losses. System availability is a common measure of operational risk [24] and thus is the *System Downtime TI* (SD). SD measures system downtimes as a rate relative to the amount of systems. SD is detailed TI, classified as *Operational:Systems*. Several studies report on statistically modelling system downtime in general [11, 123]. Due to the capabilities of parametrisation, a Gamma distribution (A.1.4) is suggested to be most flexible in modelling downtimes. Again, as the last of the three Operational TIs, SD is equally relevant to both of the regarded VO scenarios, even slightly more so to the telco scenario. This short lived VO application domain heavily relies on its IT infrastructure, the downtime of a crucial (set of) system(s) in the home operator's domain, e.g. a registry or naming service, endangers achieving the VO's business goal, serving the customer. Aerospace scenarios in contrast suffer as well from system downtimes, but more time is available to carry out an alternative work plan or simply deal with the incurring delay. A system downtime in that context may be the reason for an occurring delivery delay, hinting at possible dependencies among the TIs.

The *Employee Fluctuation Rate TI* (EF) captures one of the most common quantitative measures for a trustee's organisational stability. EF reflects the contentedness, organizational climate and labour oriented perspectives from an employee's perspective. It measures the rate with which employees enter and leave the company. The employees of an organisation are supposed to be the best ones to know about the overall organisational trustworthiness of their employer. EF counts as a detailed TI as well while still requiring interpretation as to the reasons, why EF exhibits changes. It is classified as *Organisational:Personnel*. [114] took the distribution assumption that EF follows a Laplace distribution (A.1.4) on the longer term. Organisations collaborating in a VO, especially those of the same role, of close geographical proximity and with similar product portfolios, share many external influences captured by different TIs. Nevertheless, such organisations may still have evolved around fundamentally different organisational structures. [126] declares the internal organisational structure to be the single most important factor that separates high performance companies from the average rest. EF captures the fluctuations in an organisation's workforce, taking an outside point of view. A gradual rise in numbers can reflect organic growth while a sudden increase may indicate a merger. More interesting especially for long lived aerospace scenarios are changes to the downside. This may reflect that know how bearing personnel, the "brains", is leaving the company for more creative work environments. From an outside point of view, there is always a certain degree of uncertainty involved in such interpretations. Again, EF becomes more relevant the longer a VO lasts.

The presented sample TIs, along with their attributes are summarised in the following Table 3.1:

Name	**Cash Flow Margin**	**Delivery Delay**	**Employee Fluctuation Rate**	**Country Bond Default Spread**	**Complaint Rate**	**System Downtime**
Abbreviation	CF	DD	EF	CBS	CR	SD
Class	Financial	Operational	Organization	External	Operational	Operational
Unit	% of CF on net sales	days	% of total employees	% on face value	% of shipped items	% of employed systems
Domain	$[-100, 100]$	$[0, \infty]$	$[-100, 100]$	$[0, 100]$	$[0, 100]$	$[0, 100]$
Δt_{update}	3 months	1 week	3 months	1 month	1 day	1 day
$\Delta t_{observation}$	2 years	6 months	2 years	5 year	2 years	6 months
Likelihood L	Lognormal	Exponential	Laplace	Student-t (normal)	Erlang-k	Gamma
Weighting ω	linear	linear	linear	linear	linear	linear
Preferences π						
very high	$[20, 100[$	$[0, 0.5[$	$[10, 100[$	$[0, 5[$	$[0, 2[$	$[0, 0.5[$
high	$[12, 20[$	$[0.5, 2[$	$[5, 10[$	$[5, 10[$	$[2, 4[$	$[0.5, 1.5[$
neutral	$[8, 12[$	$[2, 2.5[$	$[0, 5[$	$[10, 15[$	$[4, 7[$	$[1.5, 3.5[$
low	$[4, 8[$	$[2.4, 4[$	$[-7.5, 0[$	$[15, 22.5[$	$[7, 10[$	$[3.5, 5.5[$
very low	$[-100, 4]$	$[4, \infty]$	$[-100, -7.5]$	$[22.5, 100]$	$[10, 100]$	$[5.5, 100]$

Tab. 3.1: List of Trust Indicators

The content of this table is equivalent to the content required for delivery of a VO application domain specific TI template. While most of the attributes remain stable even across VO application domains - a TI's name and distribution assumption is not very likely to change - the trust preference mapping should be carefully adapted by a business expert. This mapping essentially captures the expert's knowledge which organisation's behaviour is deemed trustworthy or untrustworthy. The mapping given here can be considered a conservative mapping that is not favouring a particular class of VO. It constitutes a starting point for being modified and adapted to fit a particular VO application scenario. When looking at EF for instance, it becomes apparent that a rate of e.g. 2 may not always be considered of average (neutral) trustworthiness. In an aerospace scenario where it is important to keep know-how carriers within the organisation and maintain a healthy expansion, it may already be considered as a measure of high trustworthiness.

3.2 Aggregation Methodology

Up to this section, only individual TIs were discussed and modeled. TIs serve as the basis for STORE's reputation measure. They model properties that inherently characterise an organisation's trustworthy behaviour. A VOM requesting a potential VO member's reputation value from STORE can not be bothered with multiple TI's distribution parameters. This would not be a meaningful answer providing trust based decision support, it would rather confuse the requestor. TIs must be aggregated towards a meaningful reputation

measure that supports VOMs in all VO classes in their decision making. Simple aggregation mechanisms based e.g. on summation or arithmetic mean, as the ones from eBay and Amazon are scalable and easy to implement, but lack in flexibility. Their highly aggregated "one value fits all" approach that makes it impossible for the requestor to express preferences what trust subjectively means to him, e.g. he would put an emphasis on trustworthy operational behaviour. A lot of effort is put in the stochastic TI model, e.g. identifying a well fitting distribution assumption or developing a VO domain adapted trust preference mapping to achieve a prediction of the TIs future evolution that is closer to reality. To retain this quality, a stochastic aggregation methodology is adopted for STORE reputation system. This methodology is based on Bayes Networks (BNs), further background information was previously introduced in section 2.2.

3.2.1 STORE's Bayesian Network

The STORE model characterises, observes and predicts each organisation's trustworthy behaviour on the basis of n different TIs. For each organisation, a BN with a three layer topology aggregates the n TIs TI_k , $k \in \{1, ..., n\}$ towards a reputation vector $\vec{R}$. The reputation vector $\vec{R} = (TC_1, TC_2, ..., TC_m, R)$ consists of m Trust Class (TC) values and a highly aggregated generalised reputation value R. For the remainder of this work, the four TCs from Section 3.1.1 are considered, Financial, Organisational, Operational and External ($m := 4$). In contrast to STORE's first model version [124, 125], the TCs complement the generalised reputation node R as top level nodes to take a VOM's own expectations of a VO member's trustworthy behaviour into account. The corresponding mechanism is described in Subsection 3.2.2. The TC of Third Party TIs is not considered since it would only lead to a recursive integration of reputation systems of the same or similar design which is not expected to gain new insight. The STORE BN is depicted in Figure 3.3 as a directed, acyclic graph (DAG). Each node holds a random variable which is listed in the node description's first line. In the following chapters, nodes are identified by the random variable's name they hold. The edges denote causal relationships from a parent to a child note. Therefore only the bottom layer leaf nodes are without parents. These relationship are encoded as conditional probabilities in the BN, denoted in the node description's second line as $P("random \quad variable"|"parent \quad node's \quad random \quad variables")$.

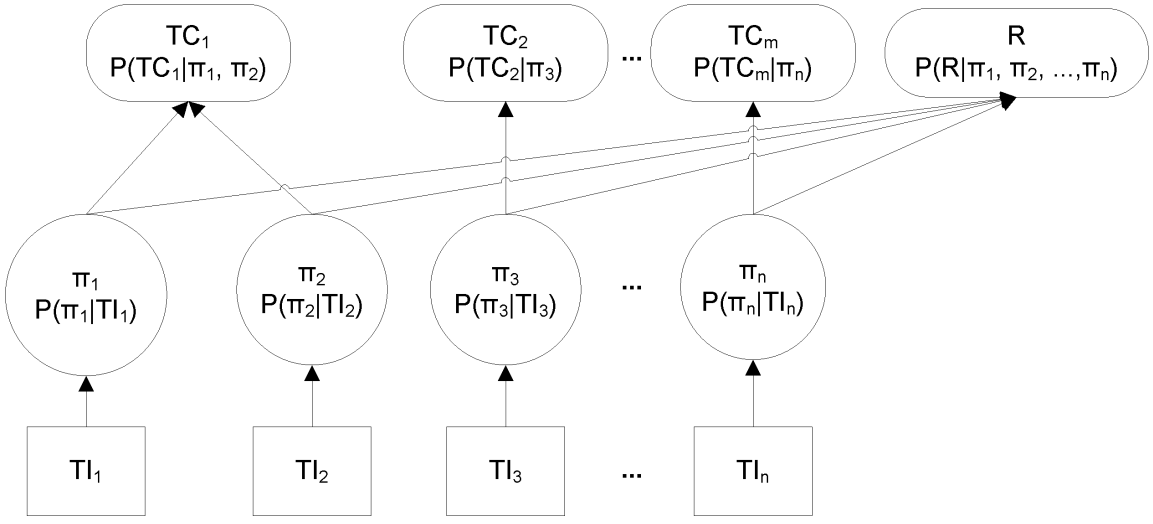

Fig. 3.3: Network Topology

The three layered BN topology is explained bottom-up, following the information flow upon availability of fresh TI data observations:

1. The bottom input layer holds information nodes TI_k, maintaining the TI random variables and their distributions. Information or TI nodes are updated with the posterior distributions resulting from the corresponding TI update 3.1.

2. The middle layer holds mediating random variables Π_k implementing the TI's trust preference mapping π_k. One mediating or preference node corresponds to exactly one TI node. TIs can be compared above this layer.

3. The topmost output layer entails the random variables corresponding to the reputation vector $\vec{R}$'s components. m nodes hold the random variables $TC_l, l \in \{1, ..., m\}$ for each TC. Each TC node TC_l's children consists only of the subset of mediating nodes Π_k belonging to those TI's within TC_l's class. The variable R represents a generalised reputation value that depends on each TI's mediating random variable Π_k in the BN.

The BN is utilised to incorporate parameter learning based on the dependencies between the TIs and their data input towards the reputation vector components. In literature, a BN's data input is frequently called evidence. The Parameter learning includes temporal forgetting of historic TI data beyond the observation time period Δt_{obs}. Since observed data is expected to be obtained from IT systems and therefore available as discrete values, the distributions maintained in the TI node's random variables are discretised. Each TI carries the states attribute S denoting in how many intervals the distribution's support is divided. All BN nodes, beginning on the bottom layer where evidence enters the BN as discrete values, do therefore not represent their random variable's distribution continuously, e.g. by maintaining the parameters of a density function's closed algebraic expression, but as discrete values in Conditional Probability Tables (CPTs). These values are the result of evaluating the conditional probabilities, denoted by the BN's edges, between a node and its parent(s). An information node holds its TI's posterior probability from the last update:

$$CPT_{TI} = P(TI) = P(\theta|X) \tag{3.2}$$

TIs have no parent nodes, therefore $P(TI)$ carry no probability conditional to any other node in the BN.

The preference nodes Π_k on the middle layer then achieve comparability of all TIs in a BN by applying a well defined trust preference mapping. It maps the heterogeneous TI states of size $s_{max}^{TI_k}$ to a uniform trust level scale of size $p_{max} \forall \Pi_k, k \in \{1, ..., n\}$. For the remainder of this work, p_{max} is set to 5, $p_{max} := 5$. An ordinal scale of size 5 to convey trust and reputation measurements is frequently used in literature [65, 103], is fine grained enough to differentiate among a sufficient amount of trust levels and has a size a requester of a reputation measure can still cope with. The connotations for each trust level are are defined as follows:

$$\begin{aligned} scale = range(\pi) = \{\ & 1 : Very\ Low\ (VL)\ trustworthiness, \\ & 2 : Low\ (L)\ trustworthiness, \\ & 3 : Neutral\ (N)\ trustworthiness, \\ & 4 : High\ (H)\ trustworthiness, \\ & 5 : Very\ High\ (VH)\ trustworthiness\} \end{aligned} \tag{3.3}$$

A preference node Π_k's CPT is computed by the following equation:

$$CPT_{\Pi_k} = P(\Pi_k = p_i | TI_k = S_j), k \in \{1, \ldots, n\}, i \in \{1, \ldots, p_{max} = 5\}, j \in \{1, \ldots, s_{max}^{TI_k}\} \tag{3.4}$$

Each preference node Π_k has exactly one parent, an information node and stochastically depends on the latter's random variable TI_k. According to the trust preference mapping, each p_i is mapped to a subset of states S_j. These subsets are disjunct and the conditional probabilities can therefore be directly computed.

The CPTs for the topmost nodes are calculated analogously. The generalised reputation node R has all of the preference nodes as its parents.

$$CPT_R = P(R = R_r|\Pi_1 = p_1, \ldots, \Pi_n = p_{max})r \in \{1, \ldots, r_{max}\} \tag{3.5}$$

R is stochastically dependent on all n Π nodes. As the previously described nodes, its support which is in $[0, 1]$ is divided into r_{max} states. For the remainder of this thesis, r_{max} is set to 100 which fits the way, a reputation measure can be inferred from the BN as described in the following section.

Similar to the reputation node, each TC node TC_l has a number of preference nodes as parents and is stochastically dependent on these. Each node TC_l has only these preference nodes as parents that mediates a TI of the top level class TC_l. For instance TC_l corresponding to the class "Operational" would have DD's preference node as its parent, but not EF's since the latter is of the TC "Organisational".

$$CPT_{TC_l} = P(TC_l = tc_o|\Pi_1 = p_1, \ldots, \Pi_{p_l} = p_{max}), l \in \{1, \ldots, 4\}, o \in \{1, \ldots, tc_{max}\} \tag{3.6}$$

with $\sum_{l=1}^{n} p_l = n$

As in the case of the R node, a TC node is divided into tc_{max} states, 100 for the remainder of this contribution. Each TC node's support is also in $[0, 1]$. The last constraint denotes that all TC nodes can not aggregate more than the n available preference nodes which are exclusively connected to one TC node.

As any other modelling method, BNs can only be as powerful as the amount of knowledge serving as model's structural input plus the available evidence. They are therefore often criticised for the subjectivity of the information they incorporate and what can be inferred from them [4]. However, in trust and reputation management, the use of subjective data is generally accepted due to the inherent subjectivity of trust itself. In this context, the BN methodology's criticism turns into an advantage.

3.2.2 Reputation Inference

In the STORE reputation system, the random variables in the top layer nodes, the TC_k and R, allow the inference of a reputation measure based on the available evidence.

To infer the trustworthiness of an organisation, the STORE reputation system the probability distribution in each of these top level nodes from the BN by setting the evidence of the TIs given by its posterior distribution $P(TI_e) = P(\theta|X)$. The latter represents the best knowledge about the current TI states S after having observed data X. The evidence is propagated through the trust mapping nodes Π and results in a particular conditional probability distributions $P(R|TI_k)$ and $P(TC_k|TI_k)$ of the reputation value. The top level nodes each hold a discrete distribution of 100 states which constitutes of too much data for a service requester. A VOM inquiring about a potential VO member's reputation can only cope with, say, 10 values or less in STORE's answer. To reach this granularity in STORE's answer, a probability distribution's basic mathematical properties are exploited. Each distribution is characterised, besides its parameters, by a centrality measure - the expectation value - and a measure of its inherent uncertainty - its variance A.1.3. The reputation vector, STORE sends as an answer to a requester is composed of the expectation values for each top level node's distribution. Optionally, as a measure for STORE's uncertainty regarding the delivered reputation vector, the requester may demand that each distribution's variance is added to the answer. The single values then become tuples which is denoted with the "[]" brackets.

$$\vec{R} := ((E(TC_1[, V(TC_1])), \ldots, (E(TC_m[, V(TC_m])), (E(R)[, V(R)])), E(.), V(.) \in [0, 1] \quad (3.7)$$

The vector components are always ordered. The TCs are an ordered set according to their index in the TI Taxonomy and the generalised reputation R comes always last. This statement is taken up again in the following chapter when STORE's runtime behaviour is presented. The order is important to allow a VOM requester to state which TCs are especially important to him with respect to a VO member in a particular VO context.

3.2.3 Learning and Forgetting

For the BN, learning is based on data observations that capture the outcome of transactions, e.g. business related transactions among VO members. This means, that if a trustee within a VO has performed the transaction a at time t_a, the trustor afterwards has the possibility to feed his experience with the other party back to the reputation service. This experience is represented by a feedback vector $\vec{r_a} \in [0, 1]$ rating that transaction. In contrast to the TI data which is raised by the system, the feedback $\vec{r_a}$ from the trustor is considered as being highly subjective. Due to this fact, STORE considers feedback about an organisation's performance only as an optional input that may improve future reputation measures for the same organisation. TIs remain the system's main source to compute reputation from.

The feedback $\vec{r_a}, r_a \in [0, 1] \quad with \quad dim(\vec{r_a}) = dim(\vec{R})$ is used to update the conditional distributions of the top level nodes. If the rater is uncertain about his previous VO member's performance in specific TCs, he may also set the corresponding feedback vector components to zero and just provide a general rating in the last component. Order and dimension of feedback and reputation vector components are the same. Unlike a TI node, the top level nodes do not possess a likelihood assumption to incorporate new evidence, a Bayesian update like in 3.1.3 is not applicable. However, an update method based on approximating the distribution of R towards the feedback value seems appropriate.

Regarding the evidence $P_e(TI)$ of the TIs at t_a, calculated by the posterior distribution of the TIs within a backward time window $[t_a - \Delta t_{obs}, t_a]$, the trust mapping nodes Π_k will also show a particular evident probability distribution $P(\Pi)$ over the preference scale. Based on r_a the system can strengthen those entries in CPT_R that refer to the evident states in $P(\Pi)$. This way for the next request under similar conditions in the TI evidence, the resulting reputation value will be closer to the former reputation feedback $\vec{r_a}$. The exact functions for strengthening is a choice of design, symmetric functions around each of $\vec{r_a}$'s components like a Gaussian or triangle function are well suited.

As the temporal update of the TIs, the reputation distributions in CPT_{TC_l} and CPT_R may have to be blurred from time to time, to implement forgetting and to counteract overfitting of the learned data. Overfitting may occur when many TIs remain the same over an extended period of time. Nearly the same evidence is put into the BN after each TI's observation period. The distributions in each node become overly "confident" in their current parameters. In particular the expectation value's probability increases and the distribution's variance decreases. If an organisation's sudden new behaviour creates the appropriate input of new, changed evidence into the BN, the distribution parameters will take a long time to change. Forgetting in BNs is basically designates procedures that approximate a probability distribution towards a uniform distribution with a certain grade and is also referred to as softening or fading. In consequence, the functions used to implement such behaviour are called fading or softening, smoothing functions. The STORE reputation system is rather resistant against overfitting and other degrading effects due to the application of a Bayes update in the information nodes. By injecting fresh data observations in a Bayesian prior distribution, the posterior distribution automatically becomes spread over the TI's support when the prior multiplied with the

likelihood distribution are normalised by the denominator's marginal probability 2.6. Such functions will thus only be applied if necessary and are not active by default.

3.3 Summary

Revisiting the the trust properties from Section 1.1, a trust model must cater for, the STORE model accounts for all of them. It considers trust relationships as a directed relationship between a VOM - the trustor - and a (potential) VO member - the trustee (trust property 1a). The model focusses on providing reputation based decision support to the VOM in selecting the most reliable VO members, being subjective from a VOM perspective (trust property 1b). Naturally, due to the asymmetric composition of a VO, one VOM may maintain trust relationships with several VO members. These are seen independent of each other. It has to be noted that such relationships are not transitive.

While most of the in section 2.1 discussed reputation models assume simple, given data or rely on binary measures such as transaction feedback, the STORE model roots trust in objective (trust property 1c), measurable, inherent properties of a trustee's organisation. The STORE reputation system can be offered as a an automated digital service in a VO environment as presented with its research prototype in Section 4.1 (trust property 1d). In that respect, it achieves comparability of trust measures across different organisation with the introduction of the preference nodes that map individual trust measures, the TIs, to a normalised, ordinal trust scale (trust property 1e). With the regular TI measurements, trust property 1f) is satisfied since freshly observed TI data captures an organisation's dynamic trust changes.

This chapter introduced the design time artefacts of the STORE reputation system - the TI Taxonomy, the detailed TI model and the TI update mechanism as well as the aggregation methodology based on Bayes Networks. Several TI examples were produced that tackled the transition to the system's runtime behaviour, especially when it comes to instantiating the artefacts, TIs in particular, and bootstrapping the system. In summary, the designtime steps to enable STORE to provide a reputation measure for an organisation, a potential VO member, are the following:

1. Select a set of TIs that describe the organisation's most interesting aspects of trustworthy behaviour.
2. Maintain the trust preference mapping for each selected TI.
3. Configure optional settings such as the system's "memory" (weighting function) or overfitting-countermeasures.
4. Deployment of previous selections and configurations into the system.

The steps are supposed to be executed by the system's owner, being a business experts in the VO application domain at hand. The first two steps are required to be started from scratch, available templates with TI sets and mappings can be (re-)used. Upon deployment of the designtime artefacts, system runtime artefacts are created and instantiated which are discussed in the following chapter.

4. THE STORE REPUTATION SYSTEM - RUNTIME

Having introduced the STORE model and designtime artefacts in the previous chapter, this one takes a look at what is required to actually instantiate these artefacts, create and maintain a running STORE reputation system that can be queried by a VOM. The chapter first presents the STORE reputation system's architecture, then illuminates the system's bootstrapping process followed by a discussion of general system aspects such as traditional security requirements and ownership.

4.1 Architecture

While being intended to be used as a centralised reputation system, the STORE architecture is designed with modularisation of components in mind. This approach not only makes an exchange and the supportability of components easier, it also facilitates a quicker integration into an existing service oriented ICT infrastructure. With the proliferation of ICT supported cross-domain collaboration as for instance desired in VOs, software design follows the Service Oriented Architecture (SOA) paradigm. It is designed consisting of smaller modules implementing a dedicated functionality in a Service Provider (SP) part. This functionality is then exposed to service consumers or service requesters via interface definitions based on open standards. This paradigm allows for quick, loosely coupled software integration, even across administrative, organisational domains. The family of Web Service (WS) standards is frequently applied which is mainly standardised by the World Wide Web Consortium (W3C) `http://www.w3.org/` and OASIS `http://www.oasis-open.org/`. Its products are based on the Extensible Markup Language (XML) `http://www.w3.org/XML/`. STORE's individual services may for instance be integrated into a VO Management System such as the one from [101] where it would seamlessly integrate its automated reputation based decision support into a web based VO management platform offering capabilities to browse for potential VO members. The platform then allows to send and maintain invitation to a subset of suitable VO members whose answers can, with added reputation measures per organisations, displayed to a VOM for a final decision. When discussing (Web) services, the following terms must be distinguished:

- The **Service Interface** is the designtime definition of which inputs the service expects from an invoking requester, which outputs it will deliver, detailed message formats and optionally further annotations and constraints. The Web Service Definition Language (WSDL) `http://www.w3.org/TR/wsdl` was created to define WS interfaces.
- A **Service Provider Implementation** entails the designtime implementation in a programming language of the function the service performs based on defined inputs to produce a defined output. Such an implementation can be a class with variables, constructors, destructors and methods in Java or .NET.
- A **Service Instance** defines a runtime artefact of Service Interface and Provider which are deployed and running in an actual system, e.g. a web container, accepting input. One service implementation consisting of interface and provider may have multiple concurrently running instances.

Using the term service implies the unit consisting of service interface and provider implementation. Figure 4.1 depicts an overview of STORE's architecture which consists of the following building blocks:

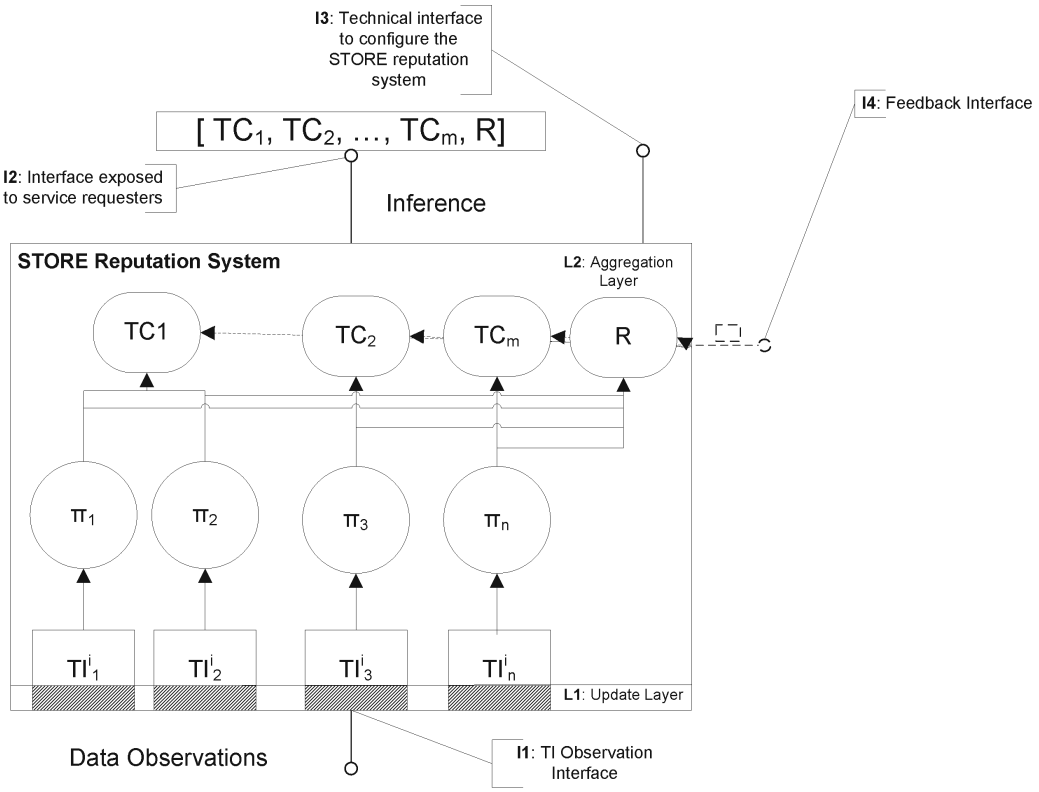

Fig. 4.1: System Architecture

Internally, STORE is divided into two layers. The bottom **Update Layer (L1)** contains the TI services. Each service maintains a TI model and its attributes from Section 3.1.2. Each TI service receives regular data observations as input. It maintains the TI specific distributions and performs the TI model update from 3.1.3 upon receipt of fresh TI data. The TI service output constitutes of the TI's posterior distribution, a time series of the posterior's random variable discretised according to the states S.
The upper **Aggregation Layer (L2)** consists of the aggregation services holding the BNs. Each observed potential VO member organisation is modelled with exactly one aggregation service instance (holding one BN) and a set of TI service instances within the STORE reputation system. The aggregation service takes the regularly updated posterior distributions from the connected TI services as input for the BN's information nodes. Each new input requires the aggregation service to recompute the conditional probabilities of every node in the BN. Figure 4.1 entails the same BN topology with the same naming convention as Figure 3.3. The aggregation service output consists of the reputation vector $\vec{R}$ from 3.7 derived from the BN's top-level TC and generalised reputation nodes.

While the interface between L1 and L2 constitutes an internal interface, the STORE reputation system also exposes interfaces to the outside, to service requesters, TI data sources and for configuration purposes. Interfaces for the latter, TI data sources and for configuration, are not supposed to be invoked by an end user, a service requester such as a VOM, and are therefore denoted as technical interfaces. First, the **TI Observation Interface (I1)** represents the defined entry point for regular TI data observations. It is a technical interface designed to be used by other applications and services and not human users. It funnels time series data for External TIs from web applications and other data providing services into the corresponding TI services, it obtains e.g. observations for confidential Financial TIs through local database services.

The **Reputation Interface (I2)** represents the main interface for human users such as VOMs inquiring about a potential VO member's reputation measure. The requester may either specify a concrete organisation or a more abstract identifier such as a required VO role he intends to inquire a reputation measure about. In the first case, a reputation vector $\vec{R}$ for the specified organisation is returned through I2, in the second case the answer consists of a list of organisations having the specified role assigned and their reputation vectors. For simplicity's sake, requests about a specified organisation are assumed for the remainder of the thesis. In addition to simply request the reputation vector, the VOM may additionally specify trust preferences, which of the TCs contribute more or less important trust aspects to the reputation measure. Trust preferences are specified in a weighting vector:

$$\vec{\omega} \in [0,1]^5, \quad \vec{\omega} := (\omega_1(TC_1), \ldots, \omega_m(TC_m), \omega_{m+1}(R)) \tag{4.1}$$

The TCs with $m = 4$ are ordered in the same manner as in Section 3.2.1. When a trust preference vector $\vec{\omega}$ is present in a reputation request, it is componentwise multiplied with the reputation vector $\vec{R}$ and the result returned. A VOM for example intending to set up a collaborative engineering VO in the aerospace domain emphasises on the VO member's environmental and financial trustworthiness, he therefore weighs External and Financial TIs higher than the others. In addition, he is reluctant to neglect other TCs entirely and includes the generalised reputation value with considerable weight. Such a trust preference vector may look like this $\vec{\omega}_{CE}^{T} = (0.5, 0.5, 0, 0, 1)$, weighting External and Financial TIs with 0.5 and the generalised reputation with 1. After componentwise multiplication with a reputation vector e.g. $\vec{R} = (0.2, 0.31, 0.8, 0.63, 0.6)$, the returned reputation measure would look like this: $\vec{\omega}_{CE} \cdot \vec{R} = (0.1, 0.155, 0, 0, 0.6)$. The VOM can decide on one glance that this potential VO member, while with a generalised reputation of 0.6 is a bit above average on the trustworthy side, is not suitable for his aerospace scenario since the reputation measures in the TCs of his interest are very low.

A further interface of the STORE reputation system is the **Configuration Interface (I3)**. It may be used by a configuration tool or a human administrator to configure technical system settings and deploy design time artefacts. This entails deploying selected TIs, adjust their trust preference mapping, configure weighting and forgetting functions, etc. It may also be used to enter initial settings, values and distributions when a new STORE system instance is bootstrapped.

Optionally, the STORE reputation system may accept direct feedback from an involved party after a transaction through the **Feedback Interface (I4)**. The feedback mechanism as described in Section 3.2.3 directly inputs a feedback vector $\vec{r_a}, \quad r_a \in [0,1] \quad with \quad dim(\vec{r_a}) = dim(\vec{R})$ compatible and with the same ordering as $\vec{R}$ into the system. As previously described, each vector component directly affects the corresponding BN's top-level node.

A first prototypical implementation of the STORE reputation system was presented in [124], revised for [125] and demonstrated at [50]. Figure 4.2 illustrates in a UML sequence diagram STORE's service

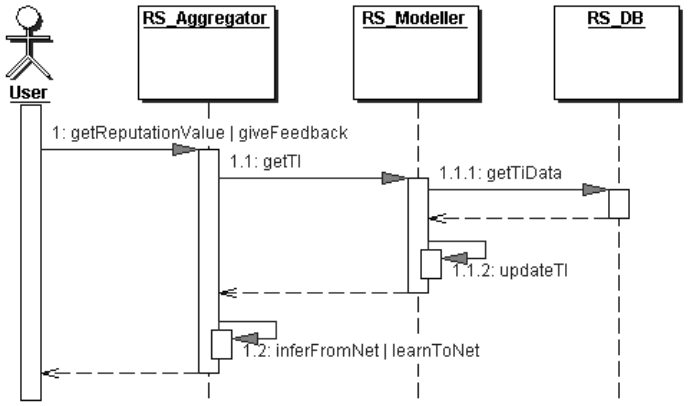

Fig. 4.2: Sequence diagram for service interaction

composition and their interactions. It depicts the interaction of a VOM (the User) with a STORE instance that models, for simplicity's sake, one VO member organisation. In this sequence, the VOM inquires about this organisation's reputation and provides feedback.

Starting bottom-up, the database service (RS_DB)[1] abstracts from individual, heterogeneous TI data sources. It can be instantiated multiple times and each instance represents the means to access one source of TI data observations. For each TI that is modelled to characterise an organisation's trustworthy behaviour, a TI service (RS_Modeller) in L1 is available. It retrieves fresh TI data observations from the corresponding database service in regular time periods (1.1.1) and updates the TI (1.2). For each modelled organisation, one aggregation service (RS_Aggregator) in L2 is instantiated. It holds and maintains the BN and obtains TI updates from a set of TI service instances (1.2). It offers the reputation interface I2 to an end user, a VOM (1) as well as the feedback interface I4. Inferring a reputation measure $\vec{R}$ from the BN and providing feedback both lead to service internal accesses (1.2) to the BN. This prototype's focus is also to explore end user interaction possibilities. It is deployed in a web container and offers a web based UI shown in Appendix A.3 for user interaction.

In the following paragraphs, the STORE reputation system is frequently identified by the reputation interface I2 it provides to its end user, the terms reputation system and reputation service are used synonymously.

A sample STORE reputation system instance may look like in Figure 4.3.

The depicted system models one organisation. Its reputation measure is based on the selection of TIs from Section 3.1.4. Each TC is represented with at least one TI, the provisioning of feedback is omitted in this example to focus on STORE's core contribution. This sample instance is revisited in Chapter 6 when it is employed for evaluation purposes.

4.2 Bootstrapping

This section discusses the aspects of bootstrapping the model and observations for a new organisation, e.g. a startup company that entered a VO application domain. These aspects are also valid when an entirely new STORE system instance is created and the first organisation is modelled. Bootstrapping such an organisational model faces two challenges, starting the TI observations without prior knowledge about an

[1] RS - Reputation System, DB - Database

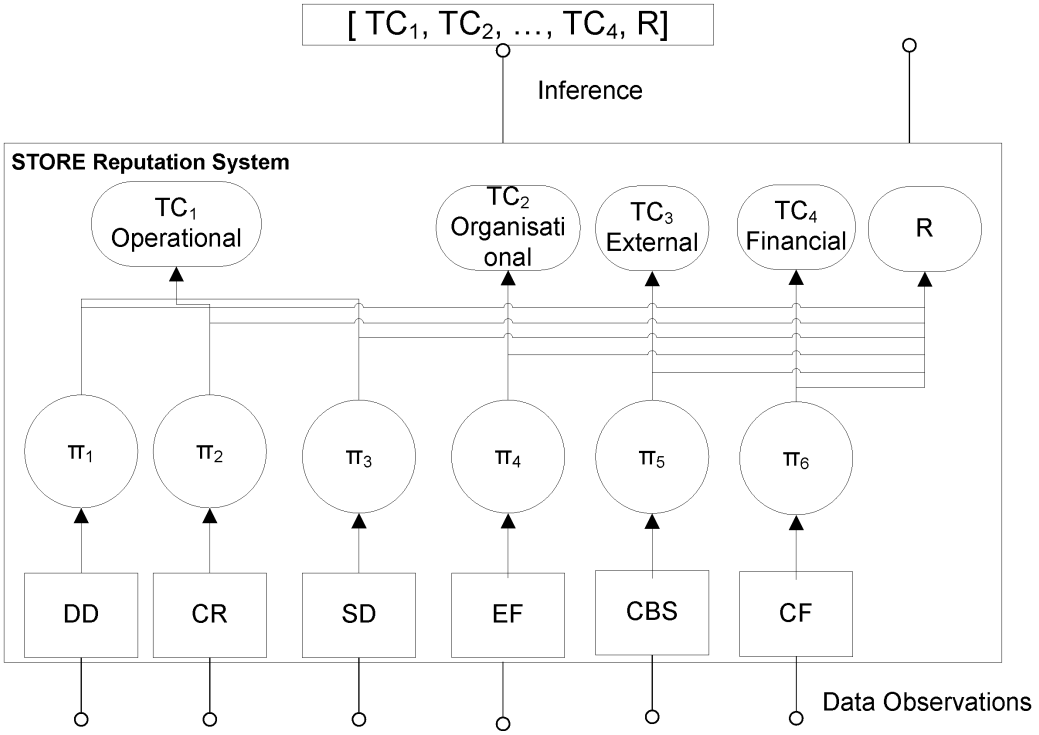

Fig. 4.3: STORE instantiation with example TIs

organisation and initialising all other BN nodes besides the TI information nodes without prior evidence from the latter. This constitutes an initial value problem.

4.2.1 TI Bootstrapping

The TI's data model as described in Section 3.1.2 mainly consists of the prior and the Likelihood distributions. Fresh data observations contribute to the prior $P(\theta)$ while the Likelihood distribution $P(X|\theta)$ is computed from historic data of the organisation's past performance. In case of an entirely new organisation there is no historic data X available and the Likelihood distribution must be initialised by other means. Having no historic data in terms of Bayesian theory means there is no epistemic knowledge about this organisation available. To make the first TI first update possible, two alternatives present themselves: Bayesian theory proposes to set the Likelihood distribution to a uniform distributed covering the entire TI domain. This constitutes an uninformed choice. The other possibility is the template based approach. If the same or another STORE reputation system instance already maintains service instances and models for other organisations from the same VO application domain as the newcomer, their historic TI data can be used to bootstrap the system. To avoid exposing confidential data, the initial TI dataset can be the result of averaging over several organisation's TI data, hereby also adding Bayesian uncertainty by cumulating the distribution's variances.

Both alternatives require a strategy with which initial trust a newcomer is allowed to enter the VO en-

vironment, if his initial TI's Likelihood distribution parameters represent a trustworthy or less trustworthy organisation. Initial setting to the former may be an incentive for certain types of attacks, such as starting over again with a high reputation measure after having committed fraud. The latter setting may inhibit a newcomer to actually be selected for a VO. This line of thought is followed in Chapter 5.

Even in case where only few epistemic knowledge is available, Bayesian uncertainty must be taken into account. More precisely, a TI state $S_\emptyset$ for which no evidence was observed results in $E(S_\emptyset) = 0$. With an expectation value of zero, the update procedure will deliver bad results, for instance produce a posterior with a zero point.

Apparently, it is a question of the state's granularity and the number of occurrences observed, how likely the occurrence of such $S_\emptyset$ becomes. Only large occurrences of $S_\emptyset$ may become a problem since they confine states with corresponding observations into increasingly smaller sets of states and hereby create an overfitting phenomenon. An established way to level out the effects of excessive granularity in a discrete distribution in general is to apply a smoothing function, for example a moving average function or an exponential smoothing over the TI states.

4.2.2 Bayes Network Bootstrapping

By addressing the TI nodes, the previous section discussed the bootstrap process of the BN's leaf nodes. This section covers the setting of remaining node's CPTs.

The mediating nodes Π_k with p_{max} states each hold the trust preference mapping based on an ordinal scale. Their CPT deterministically incorporates the mapping from the TI states S to the preference $\pi(S) = p$ and can be defined in a straight forward manner by

$$CPT_\Pi = P(\Pi = p|TI = S_s) = \begin{cases} 1 & \pi(s) = p \\ 0 & else \end{cases} \tag{4.2}$$

This bootstrapping function yields a result only when a trust preference interval p matches a state S_s it is mapped to.

The actual aggregation of the trust indicators is incorporated in the CPTs of the BN's top-level nodes, the TC and the reputation node R. In order to obtain meaningful reputation measures from a newly started STORE instance even before the first TI update occurred, an initial correlation between the top-level nodes and the trust indicators TI by means of the preference nodes Π can be assumed. This assumption holds due to the direct stochastic dependency of TI and mediating nodes, as well as the latter and the top-level nodes. Both combined lead to the assumed correlation by means of variable elimination where the mediating nodes random variables are marginalised.

A higher TI value on the preference scale also refers to a higher reputation. When dividing the domain of a top-level node, say R, $dom(R) = [0, 1]$ into p_{max} equidistant intervals R_r , $r \in \{1, ..., p_{max}\}$ likewise representing the states of R, we can cast this correlation into a single conditional probability function $P(R = R_r|\Pi = p)$ that has its maximum where $r = p$, that is where R falls in the preference interval given by Π.

As a first simple approach, a two-sided linear function normalized by its sum in the denominator is proposed for $P(R|\Pi)$ with only a single mediating node Π. For a better visual understanding, this function is plotted not statewise, but in a continuous manner in 4.4 applying different preference values.

$$P(R|\Pi) = P(R = R_r|\Pi = p) = \frac{p_{max} - |p - r|}{\sum_{r=1}^{p_{max}} p_{max} - |p - r|} \tag{4.3}$$

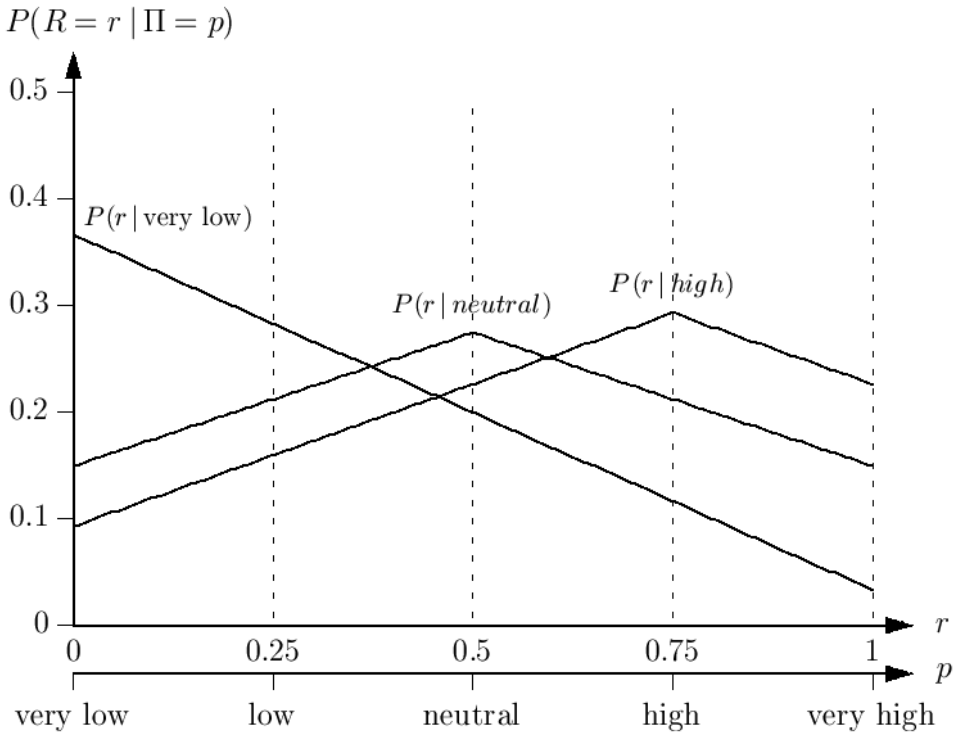

Fig. 4.4: Bootstrap function for single conditional probability $P(R|\Pi)$

In Figure 4.4, two horizontal axis are shown. The lower one depicts the trust preference scale with $p_{max} = 5$ and the intervals interpretation. The upper one shows the states r of the reputation node R. The vertical axis plots the conditional probability $P(R = r|\Pi = p)$. In this linear approach, the bootstrap functions reach their maximum when r matches a given p. The latter determines an organisation's initial trust level.

For the purpose of bootstrapping the TI's stochastic independence is used. Extending the approach to n trust mapping nodes $\Pi_k, k \in \{1, \ldots, n\}$ are then aggregated by defining the generalised reputation node's CPT as the joint conditional probability depending on all preference nodes Π. Accordingly CPT_R is composed by computing and normalizing the product of the single conditional probabilities $P(R|\Pi_k)$, using the TI's stochastic independence.

$$CPT_R = P(R = R_r|\Pi_1 = p_1, \ldots, \Pi_m = p_m) = \frac{\prod_{k=1}^{m} P(R|\Pi_k)}{\sum_{r=1}^{p_{max}} \prod_{k=1}^{m} P(R|\Pi_k)} \tag{4.4}$$

Note that the CPT_R stores a $n + 1$ dimensional function with total ${p_{max}}^{n+1}$ entries, one for each combination of input and R states. The resulting distribution in R is plotted in 4.5 for a combination of two different preference nodes. It becomes apparent that generally $P(R|\Pi_1, \Pi_2)$ has its maximum at the average of the two preference values (i.e. in this case in the middle), whereas the kurtosis (or "peakedness") of the distribution depends on the distance between p_1 and p_2.

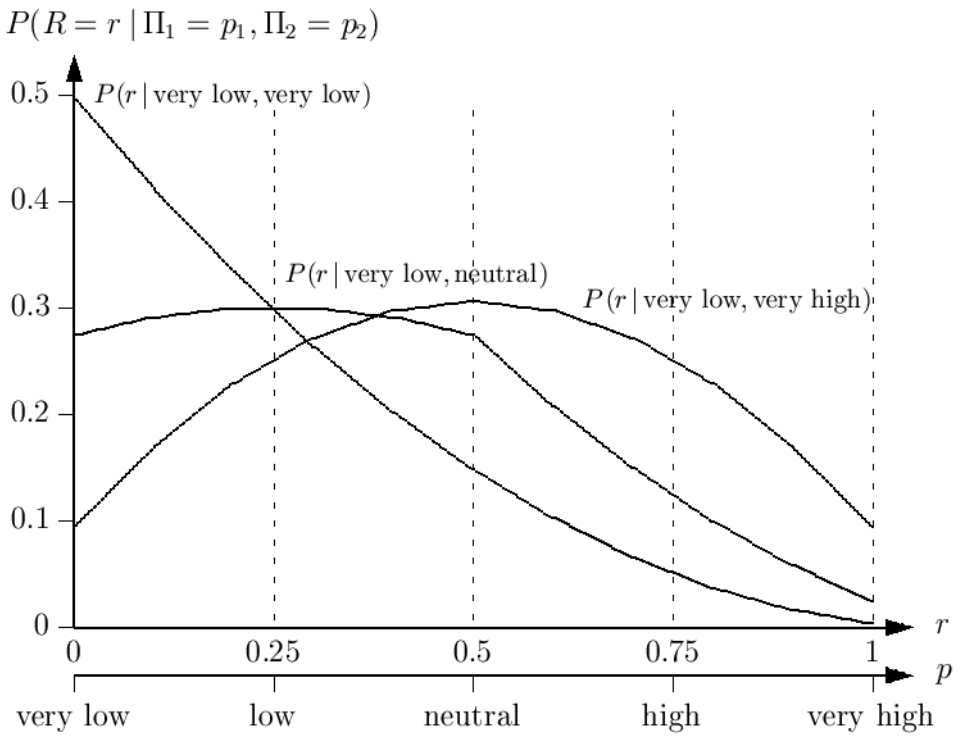

Fig. 4.5: Bootstrap function example $P(R|\Pi_1, \Pi_2)$ with two mediating nodes

The axis plot the same variables as in Figure 4.5, the vertical axis' conditional probability in this example is extended to two dependent mediating node's random variables Π. This last example only dealt with two of the n present mediating nodes for n TI nodes. In general, the reputation node R is bootstrapped taking all n mediating nodes Π into account. The TC nodes are bootstrapped analogously, each TC node can only be bootstrapped its parent mediating nodes that hold the trust preference mapping for the TI nodes belonging to its trust class.

4.3 System Aspects

The previous subsections presented STORE's architecture and related runtime aspects such as the steps to bootstrap a system instance. When maintaining a STORE system, the owner has to consider other, mainly non-functional, requirements as well. The following subsections will therefore discuss aspects of the system's ownership itself, as well as information security and availability requirements.

4.3.1 System Ownership

Offering reputation measures of organisations for decision support purposes unavoidably raises the question, how trustworthy the source of these measures actually is. Distributed reputation systems, with no central point to query, therefore frequently take a trust measure of the sending party or system node along with the requested reputation measure into account. In case of the centralised STORE system, the question precisely

targets the trustworthiness of the system's owner or hoster. In previous discussions - departing from Subsection 3.1 - the owner was assumed to be an explicitly Trusted Third Party (TTP). Taking the global economy's consolidation, specialisation and companies focussing on their core competencies into account, this assumption especially applies to outsourcing environments of specialised Application Service Provider (ASP) or IT hosting. In particular in the aerospace scenario's case, this specialisation and consolidation becomes apparent when looking at the required roles. A design specialist for instance focuses his core competencies on simulating plane designs while storage of these design data is kept with another specialised organisation. Abstracting from this case to application and IT outsourcing to a hoster in general, this hoster already is considered trustworthy enough to manage an organisation's confidential, even financial and organisational, data along with maintaining the platform running the organisation's business processes within the hoster's administrative domain. And more, an ASP's business model is the parallel hosting of applications from a large amount of customers which may even be fierce competitors. If one hoster is entrusted by multiple organisation's with (parts) of their core assets, he becomes a TTP and becomes a candidate to host a reputation system such as STORE for his customer domain.

4.3.2 System Security

By providing automated, reputation based decision support for collaborative environments such as VOs, the STORE reputation system implements a "soft" security mechanism. This mechanism reduces the risk of a VOM selecting a "lemon" [6], an untrustworthy VO member. It is not designed to replace "hard" security addressing traditional security requirements. Beyond the VO's formation phase, when the operational phase starts, the integrated ICT infrastructures have to take care of application and organisation specific authentication, confidentiality, integrity, and non-repudiation.

This also hold true for the STORE reputation system itself. Depending on the business model, if the system owner charges for individual reputation queries or for querying potential VO member reputation measures only within a specific application domain, the exchanged message contents may pose confidentiality requirements to a varying degree. Even from usage patterns, which VOM queries at which point in time, an eavesdropper may infer confidential information that he can use to his advantage. Actively altering the system's response by injecting bogus reputation measures may disrupt beginning fruitful VO relationships - a data integrity requirement.

State of the art security standards and mechanisms can be applied to secure communication between a user and the STORE reputation system. Since STORE is designed to be deployed in a SOA environment these are typical web security standards. Confidentiality and integrity requirements can be met on a communication channel basis with a sensibly configured Transport Layer Security (TLS) protocol on the transport layer [2]. Meeting these requirements on message level is also possible, e.g. by using the Web Service Security (WSS) standard [3]. All these mechanisms hinge on authenticating the communicating entities in this VO landscape. While authentication of the STORE system itself can be supported by a server certificate issued by the (trusted) system's owner or another trusted agency, authenticating the requester invoking the reputation interface is more difficult. This would be the VOM of an emerging VO. VOM hereby represents an organisation or more precisely the role an organisation will play in this VO. To authenticate the requester, a principal, e.g. a contact person for this VO in the VOM's organisation needs to have been provisioned, after verifying his identity, with authentication credentials trusted by the STORE reputation system. This

[2] `http://www.ietf.org/html.charters/tls-charter.html`
[3] `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss`

implies, STORE or its owner maintains an identity database of VOMs allowed to access the system. This requirement can be weakened to persistent pseudonyms [41]. Keeping track of identities is also important if the (optional) feedback interface is offered. Feedback always relates to a past business transaction within a particular VO context. Besides this temporal and VO aspect, it can only be given by a qualified principal. Since the STORE reputation system observes (potential) VO members, the VOM is the only qualified principal to judge the VO member's performance during this past transaction and if it matched his expectancy of trustworthy behaviour as raised by the received reputation measure. Therefore, his identity needs to be verified prior to accepting his feedback which must be only counted once. Other security aspects target STORE's reputation model and mechanism directly. These are considered attacks on the reputation system and its community itself and are discussed in the following Chapter 5.

4.3.3 Availability and Scalability

Two availability requirements are paramount in the STORE architecture. First, data availability of TI observation data has to be ensured. While some TI can be observed from public sources such as CBS, others are kept strictly confidential by the organisation that is observed. EF is an example for the latter case. The Bayesian update mechanism, in particular using a posterior distribution for each TI, can cope with intermittently missing data to a certain extent, but not its entire unavailability. Exposing such data, besides being compliant to legal requirements, again mean trusting an external party. Revisiting the ownership discussion reveals two possible organisational setups for such exposure. In an ASP and IT outsourcing environment, where the hoster acts as a TTP, organisations may decide exposing such data would be in their advantage. This data is never exposed in raw, detailed form affiliated with their organisation, but only leaves the STORE system as a highly aggregated measure within the reputation vector. One organisation following this path would carry a more detailed reputation measure within the STORE system which provides an improved decision support to a VOM. Provided the measure represents a high reputation, the VOM will select this organisation with a higher probability, motivating other organisations to follow this example. Assuming this takes place in a large, homogeneously administered hosting environment, obtaining TI data observations can happen with low technical effort. The other possible setup follows the approach of rating agencies like Dun&Bradstreet. A dedicated TTP, also hosting the reputation system, obtains such confidential data due to proprietary and laboriously negotiated agreements. This approach comes with a lot of manual and administrative effort.

The second availability requirements targets the system's availability itself. The STORE system architecture, due to its modular design, scales very well. Most functionality with the exception of the aggregation service scales linearly, TI Services can be instantiated on separate machines, can be migrated comparably easily due to the loose coupling of services in a SOA environment. The exception, the aggregation service, maintains the BN of an observed organisation. The STORE model explicitly explores the possibility of a rich data model of trust properties with included semantics, meaning which aspects of trusting behaviour this property characterises. On top, to allow for better predictions, a stochastic instead of a deterministic modelling approach is followed. Therefore, with each TI update, the entire nodes in the net must be recomputed. Having modelled TIs of a high update frequency, this may lead to a high load for the IT infrastructure, CPU and main memory consumption in particular. This can be mitigated to a certain extent by carefully selecting the algorithms applied to (re-)compute the BN, see also 2.2. The naive approach using internal variable elimination is the most inefficient. Others exploit localisation properties that for some TIs only requires partial net updates while another class uses graph transformation algorithms, transforming the BN topology into a simpler one requiring less computations per update. Hybrid approaches combining these exist as well

and are documented in [72, 85, 77, 63, 115].

4.4 Summary

This chapter tackled the topics of an operational STORE system instance. First, the STORE architecture was described. The architecture is made up of modular components that can be encapsulated as three well defined, reusable service implementations. The interface descriptions and service interactions were also provided. Special care was taken to discuss the bootstap process when setting up a new STORE reputation system instance. The bootstrap process addresses all nodes on every layer in the BN. Several bootstrap approaches ranging from pure value assignments to the integration of an already pre-learned BN template, e.g. from an existing STORE reputation system instance of the same VO application domain, were suggested. This chapter concluded with a discussion of system aspects such as ownership and security. When assigning initial trust and reputation values to newly joined organisations, a careless assignment may give rise to attacks on the reputation mechanism itself. Such attacks and others are the topic of the following chapter that will introduce an attacker model and classify attacks on reputation systems in general. Each attack is analysed and its applicability on STORE in particular is assessed. Finally mitigating measures are suggested for each attack.

5. ATTACK CLASSIFICATION AND MITIGATION STRATEGIES

As for all information systems, the possibility of an attack on a reputation system can not be ignored [29]. Since an online reputation system such as STORE must be available to requesters within a larger network, e.g. the Internet, attacks can not be fully excluded but can be mitigated with countermeasures. In a VO application scenario, the reputation system is designed to be available to a VOM intending to query about the reputation of potential VO members. To deal with potential attacks on the STORE system, first, the involved actors and their roles, as well as the vector of such an attack must be analysed and understood. Second, mitigating countermeasures can be recommended. This chapter provides a threat model and a classification of possible attacks on the STORE reputation systems. For each identified attack, a set of mitigating countermeasures is recommended and finally the results are summarised in two tables. It also elaborates on the actors and their roles who benefit from mounting such attacks. Denial of Service (DoS), communication eavesdropping and other external, system level attacks are not subject of this chapter, but were discussed in Section 4.3.

5.1 The Actors

A VO setting or community entails the following three VO actor roles. The attacker, reaping a benefit from manipulating the VO environment or otherwise tampering with the STORE reputation system is expected to impersonate these roles.

The Service Provider
The Service Provider (SP) offers a service based on his specialised expertise and core competencies to other VO members. A SP typically enacts the role of a ”VO member”.

The Service Consumer
The Service Consumer (SC) invokes and consumes a SP’s service. Section 2.3.1 described the two contrasting telco and aerospace scenarios in detail and how the VOM coordinates and integrates processes during the VO’s lifecycle phases. Therefore, the role of a ”VOM” consumes the VO member’s, the SP’s, services.

The reputation system hoster
The STORE reputation system’s hoster, also abbreviated as the hoster, owns and maintains the system instance that provides reputation based decision support in particular VO scenarios. The hoster is, as in previous chapters, assumed to be a TTP if not stated otherwise.

VOs do not prosper in anonymity. When its members are selected at the end of the formation phase, one or more legally binding contracts are signed by all collaborating principals, clearly stating each one’s duties throughout the VOs, constraints and penalties in case of breached contract clauses. A legally binding

document requires the identity verification of each signing party. The principals, VOM and VO members, are rather companies and similar organisations than individuals. Organisations leave a larger footprint in the real world than individuals and are less likely to remain anonymous or commit identity theft in order to fake their identity. These facts decrease the likelihood of schemes from external attackers, besides the previously discussed system-level attacks. These can be addressed by establishing properly configured hard security measures. Attacking the soft security measure, the reputation system itself, will most likely be an attack from within the community the system serves. Therefore, the role of an *attacker* exists but will be assumed by a principal enacting one of the three roles from above.

5.2 Attack classes

Considering which VO community role an attacker assumes means devising an attacker-oriented threat model. An attacker may assume one of the three roles of a SP, SC or reputation system hoster. The trust model implied in this section generally perceives the role assumed by the attacker as not trusted while the other two roles are trusted. When analysing the individual attacks, the attacker's assumed role is specified in the beginning. Certain attack classes, such as collusion attacks, require a modified trust model. There, principals enacting different roles, for instance SP and SC, cooperate in an attack and therefore both roles are not trusted. In each of these cases, the trust model is explicitely stated.
The following paragraphs enumerate possible attacks on the STORE reputation system, their description tackle the attacker's motivation to mount such an attack and outline his benefit and gain. None of these attacks is specific to only one VO application domain or specific application scenario. While most of these attacks are known from literature and were researched in more general contexts, e.g. of online communities, some are new and can be exploited in VO environments. These attacks are highlighted in the description. Some attacks can be mounted by colluding attackers assuming different roles as their attack vector. Countermeasures mitigating these attacks are provided inline in the corresponding description. Table 5.1 summarises the attacks from this section, Table 5.2 the countermeasures.

5.2.1 Attacker enacts SP role

The attacks in this section are prone to be exploited by an attacker assuming the role of a SP in the context of the STORE reputation system. Each attack is numbered for later reference. Many of these attacks exploit feedback based reputation systems. These attacks are of less interest for discussing possible attacks on the STORE reputation system, but since STORE offers an optional feedback mechanism, they are enumerated nevertheless.

A1 - Identity Spoofing

Spoofing means to hoax, trick or deceive. Identity spoofing refers to an attacker faking or assuming another principals identity with the intent to deceive others in believing the attacker is the spoofed principal. This attack typically serves as a staging attack, a prerequisite to mount a secondary attack to achieve the desired gain. Spoofing one's identity requires an attacker to obtain certain attributes defining his target's identity. In an online community such as a social network, this may be a user name. Furthermore, attributes such as authentication credentials, e.g. a password, to exploit the spoofed identity may be required to be obtained as well. While identity spoofing in online communities of individuals may lead to an immediate gain, e.g. by committing credit card fraud under the spoofed identity, it is less likely to succeed in VO environments. VO environments rarely offer such an immediate gain, instead, an attacker must sustain the cover of an

assumed identity over a longer period of time, especially in long-lived VOs. Identity spoofing is mitigated by traditional security measures, a sound identity management complemented with strong authentication mechanisms[41, 96]. Already due to legal requirements, VO members - the SPs - enter legal relationships by signing contracts during the formation phase, member identities are known and verified. Furthermore, organisations are more likely to participate in a VO than individuals. An attacker aiming at launching a secondary attack is required to invest more resources to spoof an organisation's identity than an individual's, though for instance a company's dedicated contact person for a particular VO would be a natural target.

A2 - Starting Over

Bootstrapping a reputation system in the beginning and managing the entry of new, previously unknown VO members to an existing community are delicate and security critical processes. Essentially, the problem the reputation system hoster faces is which initial reputation values are assigned to new, unknown organisations. Low values inhibit the acceptance of newcomers which are unjustly singled out and never selected for VOs. High values provide an incentive for "starting over"[41]. A new potential VO member with initially assigned high reputation values may decide to cheat or simply provide bad service. When he is no longer selected by VOMs for VOs due to decreasing reputation values, he leaves and re-enters the community under a new identity. The attackers gain typically centres around maximising their own profit with bad, low quality service while participating in VOs.

As with A1, starting over is easier to conduct in online communities of individuals than in VO environments consisting of organisations. A2 can not be ruled out due to low likelihood of appearance though. More dynamic VO environments such as the high-tech industry are used to a high (and increasing) turnover of specialised supplier companies. Especially with emerging products and potentially high profit margins, starting over may be worth an attacker's while even when each attack requires founding e.g. a new start-up firm and with VO lifetimes of months to several years.

Again a rigid identity management that allows the reputation system to track identities behind an organisation's facade delivers a mitigating measure. Since the lifecycle of identities is of importance to thwart this attack, persistent pseudonyms still allow the tracking of identities behind the pseudonyms and offer a higher level of privacy. It must be ensured though, when issuing pseudonyms, that these are persistently tracked and no duplicates are handed out to the same identity. The STORE architecture itself provides a certain protective measure against starting over, by taking the dynamics of trust into account. TIs characterise an organisation's trustworthy behaviour and their measurements are aggregated towards a reputation vector. Since TIs are observed periodically, even throughout the VOs operational phase when an attacker's low quality service level shows, STORE captures this change in behaviour and a VOM can either invoke the reputation service again or may be notified. He can then decide to take action, negotiate with the misperforming VO member or in the worst case replace him. Promising, but still experimental incentive mechanisms were proposed to help with a reputation system's bootstrap problems and also mitigate effects such as "starting over". Reputation lending [45] looks especially promising since the authors claim, the mechanism can be used in conjunction with any reputation system. The mechanism design is inspired from real life where newcomers try to convince a local introducer, a "mentor", to introduce them to a community. The introducer hereby lends the newcomer of his own reputation. If the introduction progresses well and a productive member of the community was introduced, the introducer receives more than his previously lend reputation back, otherwise it is lost (an incentive for introducers to be risk aware). This mechanism and its design fits a VO environment with the STORE reputation system very well.

A3 - Freeriding

Freeriding refers to a subset of principals in a community that benefit unjustly from the well performing remainder of community members. Freeriding is usually less of a problem for centralised reputation systems, but more so for decentralised ones. While a centralised system is better able to identify SPs offering lower service qualities and distinguish them from others, freeriders have a higher chance to hide in decentralised structures. They gradually gain a higher profit by providing a worse service quality than the honest SPs who ensure a healthy environment based on good service. A typical example of freeriding occurs in Peer-To-Peer (P2P) networks for sharing files, e.g. using the Gnutella protocol, where a healthy environment requires participants who download a file from several SPs to also share the already available file part. Freeriders will not share and just download hereby decreasing the availability of files by saving their own bandwidth [3].
Freeriding among SPs may also occur in cases when a STORE reputation system instance is used to compute reputation values for potential VO members of several VO application domains. A freerider would be a potential VO member who is able to enact business roles, e.g. as a storage provider, in more than one VO application domain, e.g. aerospace and telco scenarios. A freerider is able to appear sufficiently trustworthy by performing well according to certain trust classes that do not require much effort and resources. A potential VO member may appear well suited for an aerospace scenario if his origin happens to be in a stable region with an up to date infrastructure (Environmental TC) and if he maintains financial stability (Financial TC). The operational trustworthiness which is the most important trust aspect in telco scenarios, may be average or slightly below. Such a SP would still be selected for aerospace scenarios and would, from a VOM's decision making perspective, superficially still appear suitable for telco scenarios. In the latter case, the freerider would reap the benefits from operational cost savings leading to average or lower service quality in telco scenarios. Such freeriding behaviour can only be successful if the attacker's reputation is not dropping too far below average, e.g. if he becomes too greedy, since he would no longer be selected for neither aerospace nor telco scenarios.

The STORE reputation system includes several architectural mitigating measures against such behaviour. Reputation values are always computed and evaluated in a VO application context. First, a business domain expert is required to provide the trust preference mapping for the mediating nodes of a particular STORE system instance. When defining this mapping, the expert needs to take the VO application domains covered by this STORE system into account. To avoid freeriding, the mapping should be more sensitive to trust changes measured by TI data of complementary TCs being of importance for the VO application domains. A skilled freerider may still stay within this level of sensitivity and can not be identified by the TC components of his reputation vector. For such cases, the generalised reputation value R is included that may still provide a means to identify the freerider by comparing the value relative to other SPs. As a third, external, factor the VOMs, the SCs, trust preferences entered as weights for each reputation vector component are taken into account when assembling the reputation service's answer. A freerider from the above example being considered as a member for a telco scenario exhibits a slightly lower value in the operational TC. The inquiring VOM would put a higher weight on this component which would scale the difference to an average or better reputation component value to a visible absolute value.

A4 - Blackmail

Blackmail constitutes an Out of Band attack that bypasses a reputation system's regular communication channels. Especially untrustworthy SPs offering low quality service use other means, e.g. mail or email, to put pressure on SCs not to report the received bad service with honest feedback. Most prominently feedback

based reputation systems, such as eBay, suffer from blackmail that can also be a prerequisite to mount other attacks such as ballot stuffing or bad mouthing where one or more SCs are pressured to provide feedback of a desired kind. Blackmail allows low quality SPs to stay in business and become selected for future business transactions despite their bad service.
The STORE system is, by design, resilient against blackmail. The reputation computation is based on observable TIs, feedback can be optionally included but is only contributing to the TI determined reputation vector. The possibility exists that a SP may be blackmailed to provide falsified TI observation data. This requires technical effort since either larger amounts of data or technical TI observation service implementations and corresponding deployed instances have to be tampered with. Such high effort reduces the likelihood of exploiting such blackmail attacks. If this attack is to be mitigated in a VO environment, the same countermeasures as in the following A11 - Observation/Feedback Tampering apply.

A5 - Negative Discrimination

Negative discrimination denotes a SP's good service offering for the majority of SCs and low quality service for a selected set of other SCs. The SP's gain lies in externalities against these victimised SCs. This attack is typically externally motivated, a SP receives for instances bribes to discriminate a select set of SC's, the competitors of another bribing SC. With feedback based reputation systems, negative discrimination is the more difficult to detect the larger the size of the community. If a SP provides his regular, best possible service quality to the majority of a large SC community, the bad feedback from the smaller discriminated subset will not significantly decrease his reputation value.
As A3, the STORE architecture is resilient against this attack since the selective low quality service is represented in the TI observations and, with an appropriate trust preference mapping, will shift the attacker's trust level significantly to the worse. Since such selective bad service will most probably affect TIs from the operational TC - the SP has no or few influence on them at a given time across SCs, a more sensitive trust preference mapping as in the case of A3 more quickly reflects behavioural changes in the reputation vector and helps to alert the reputation system hoster of the attack.

A6 - Positive Discrimination

A6 is nearly the opposite of A5. Instead of providing low quality service to a select set of SCs, the SP offers exceptionally good service quality to these and his regular one to the majority of the SC community. While initially sounding counter-intuitive, a malicious service provider benefits from this attack by possibly triggering a ballot stuffing (A9) effect. The select set of favoured SCs, e.g. from a particular VO application domain, will provide positive feedback for this SP furthering his - tactical or strategic - goals in this application domain. STORE's resilience arguments and the others from A5 also hold in this case, since mainly feedback based reputation systems are susceptible against this attack.
A5 and A6 are prone to be mounted by a group of attackers and executed in a concerted action. Executed not by a single but by a group of attackers classifies as a collusion attack, multiplying the attack's effect at least with the number of colluding attackers. In consequence, the group of attackers benefits from the added gain. If, for example, A6 is launched with the intent of triggering a ballot stuffing effect, more attackers aiming at the same target group of SCs can achieve a larger amount of the desired feedback which is submitted for the respective participating attackers from the group. In a VO environment, it is important that a SC is able to identify SPs fitting desired business roles and inquire about their trustworthiness for decision making purposes. It is not important if SPs can identify each other, it would even give rise to the attacks A5 and A6, since malicious SPs would be able to select their victims. In this case, a controlled anonymity can further mitigate these attacks. This can for instance be achieved by creating (persistent) pseudonyms identifying

a SP when he registers to be observed by the STORE reputation system[29]. Real identities are still used when a SC inquires about a SP's reputation, but only pseudonyms are used among SPs if there is a need at all to be identifiable through the STORE reputation system e.g. at a VO's formation phase.

5.2.2 Attacker enacts SC role

The following attacks require an attacker to assume the role of a SC. All of these attacks focus on a feedback mechanism and are only relevant within the STORE context, if the optional feedback mechanism is activated. For each attack, a general mitigation measure is suggested that also applies to the STORE reputation system with activated feedback mechanism.

A7 - Feedback Starvation

Feedback starvation assumes an attacker actively tampering with the feedback mechanism. It refers to an attacker playing the role of a SC providing no feedback at all or only biased feedback. The latter case is more interesting, the SC may be the actual attacker pursuing his own goals. For instance when providing bad feedback for a SP delivering high quality service may lead to reduced fees for future business transactions of retained high service quality. But the SC may in this case also be the attacker's (acting as a SP) victim who first mounted A4, a blackmail attack, to coerce the SC to provide positive feedback after having participated in a bad business transaction with the attacker. SCs are also likely to provide biased feedback out of fear for retaliation if the SP has a channel to react upon such feedback, lashing out at its source.
No feedback being provided can be detected when coupling the feedback to the business transaction it belongs to, e.g. by allowing only the participating SC from a transaction identified by a unique ID to provide feedback once [49, 29]. Biased feedback can at least be detected using collaborative filtering techniques[29]. Filtering to detect A7 is best based on data, the feedback data itself, identifying suspicious instances based on a particular SC habit of giving feedback for a certain type of business transaction and with a SP offering a similar service quality. The SP can be assessed based on other SC's feedback. Incentive mechanisms stimulating the submission of honest, regular feedback can further mitigate this attack[96].

A8 - Feedback Flooding

Feedback flooding is mounted by an attacker, e.g. as a DoS attack, to destabilise the feedback mechanism or reduce the reputation system's availability. The attacker or group of attackers send large amounts of unrelated feedback data.
Mitigating measures to prevent A8 are to the largest extent the measures ensuring system availability discussed in Section 4.3. The feedback mechanism must ensure that only the actors participating in a business transaction are allowed to submit feedback about its outcome and only once. Measures to detect feedback flooding, e.g. if a design flaw in the feedback mechanism can be exploited, come again from collaborative filtering research. With A8, frequency based filtering may help, detecting for instance an unusual large amount of unrelated feedback originating from an actor.

A9 - Ballot Stuffing

Ballot stuffing is similar to A9, but is directed and goal oriented. The attacker or group of attackers send large amount of legitimately looking feedback to boost the reputation of single or group targets. The same mitigating measures as in the case of A8 apply. In addition, since ballot stuffing aims at boosting reputation, collaborative filtering based on feedback data can be applied besides frequency filtering. A sudden increase

of positive feedback for one or a group of targets can be detected as a bias. Hybrid approaches based on data and frequency filtering combine the advantages of both approaches[29].

A10 - Badmouthing

Badmouthing differs from A9 only with respect to the attack's goal, instead of increasing the reputation of a single or group of targets, it aims at decreasing this reputation. This is achieved by sending negative feedback. The same mitigation measures as in the case of A9 apply.

5.2.3 Attacker enacts hoster role

Up until now, throughout this thesis and in literature, the party owning and hosting a central reputation system was considered as explicitly trusted, a TTP. This assumption can be taken in organisational structures with directly communicating parties. In a global economy, organisations increasingly focus on their core competencies. Other specialised expertise which they do not have but which is required to deliver a service or manufacture a product is bought in from other organisations. Taking the example of an aerospace scenario, a company assembling the interior of a passenger plane buys in specialised labour, for instance the manufacturing of passenger seats, as cheaply as possible to stay competitive. Organisation from countries with low ancillary labour costs, e.g. China or India, deliver cheapest and are selected which is the motivation for outsourcing to such countries. Effects such as outsourcing deepen cross-organisational structures, companies such as the one responsible for the hull's interior are visible in VO communication patterns, but not necessarily their subcontractors, e.g. the seat manufacturer. The former acts as a communication proxy for the latter. Both parties may carry greatly differing reputation vectors which is not necessarily a result of conscious and planned behaviour. A Chinese manufacturer for instance may be physically located in a region of low quality infrastructure and a high likelihood for natural disasters which lowers his value for the environmental TC. The same holds true for the hoster of a reputation system. From an outside perspective, this principal exposes the reputation service, answers reputation queries and acts as a single point of contact for SCs. Behind the scenes, data acquisition, obtaining and handling TI observation data, may be outsourced to a specialised analytics firm and the reputation system instances may be physically maintained by an IT outsourcing company offering cheap datacenter services. The latter outsourcing firms may not be as trustworthy as the official reputation system owner, justifying this line of thought[69]. These arguments are especially relevant in cases when not looking at simple flat communities with bilateral communication and transaction patterns, e.g. webshops or social networks, but when more complex organisational structures and business transactions are common as in a VO environment. The STORE reputation system is designed to be employed in such environments. Dealing with potentially untrustworthy or even malicious reputation system hosters guises itself as rather difficult, but if this is identified as a risk in certain security critical application domains, e.g. a VO answering a military tender. The following mitigating measures on technical and legal level can be suggested.

A11 - Observation/Feedback Tampering

The principal actually running the physical reputation system instance has full access to all data that is aggregated towards an organisation's reputation value. He can therefore alter and manipulate this data that encompasses the TI observations as well as feedback if this mechanism is used, which leads to a false reputation results. The motivation for such action may be external, a SP may use blackmail to boosts his reputation in this way. If bribery from a SP's side is applied this would constitute a collusion attack between SP and reputation system hoster. A hoster may also be motivated to mount such an attack on his own with

the prospect of financial benefit, e.g. on the stock market, if his tampering has an effect on the victim's share price.
A malicious reputation system hoster has the potential to destroy existing trust relationships in even large business communities and hinder the emergence of new ones. Misusing this responsibility should be severely punished e.g. by enumerating strong punishment clauses in the legally binding document, e.g. SLA, guarding the use of the reputation service. Since such misuse is difficult to detect from the outside, for high-security application domains technical measures such as the application of secure multi-party computation and other privacy preserving computation measures[5, 21] are suggested. Such measures exploit the homomorphic properties of elaborate cryptographic mechanisms and protocols. Potential VO member's TI sources yield no longer plain TI data observations, but the data is encrypted according to a certain scheme. This scheme is designed with a homomorphic[1] property with respect to one or more operations in that applying this operation to the encrypted data leads to a result that, decrypted, is the same as if the operation would have been applied to the plain data. In STORE's case, the scheme would be designed to be homomorphic with respect to the operations used to compute the reputation vector from TI data observations, which is essentially the Bayes update from Equation (2.4) used to inject fresh TI observations into the BN and which is also the basis to evaluate conditional probabilities along the BN's edges. Secure Multi-Party Computing has the disadvantage of adding additional overhead for performing cryptographic operations and one design for a particular application, data and operation, can usually not flexibly adapted to other applications since the homomorphic property breaks due to other application specific operations.

A8 - Feedback Flooding

Feedback flooding in this section differs from the attack in the previous section only in the role, the attacker assumes. A malicious reputation system hoster may decide to flood his competitor's feedback mechanism with false or duplicate feedback to cripple his service and lower the system availability. In turn, he might gain new customers. However, mitigating measure do not differ, the same ones enumerated in the previous section apply.

[1] see definition in Section A.1.1

Actor/Role	A1 - Identity Spoofing	A2 - Starting Over	A3 - Freeriding	A4 - Blackmail	A5 - Negative Discrimination	A6 - Positive Discrimination	A7 - Feedback starvation	A8 - Feedback flooding	A9 - Ballot stuffing	A10 - Badmouthing	A11 - Observation/ Feedback tampering
Service Provider (SP)	claim SP identity of higher reputation	SP exploits high initial reputation value with bad service and then starts over with a new identity	SP provides intentionally about average service, just to retain enough reputation to stay in business while other SPs do their best to support the environment with high quality service	SP threatens SC to avoid bad feedback after delivering low quality service	average service provided except for few buyers to gain externalities against these victims	very good service for a set of SCs, e.g. to trigger a "ballot stuffing" effect	n/a	n/a	n/a	n/a	n/a
Service Consumer (SC)	n/a	n/a	n/a	n/a	n/a	n/a	no feedback at all or biased feedback provided	large amounts of unrelated, duplicate or false feedback is provided without clear intention, e.g. to mount a DoS attack	provisioning of large amounts of feedback with the intention to boost a SP's reputation	provisioning of large amounts of feedback with the intention to damage a SP's reputation	n/a

Tab. 5.1: Classification of Attacks on Reputation Systems – continued on next page

Trusted Third Party (Reputation System Hoster/ Owner)	n/a	n/a	n/a	n/a	n/a	n/a	n/a	a malicious hoster sends large amounts of unrelated feedback to flood a competitor's feedback mechanism	n/a	n/a	an untrustworthy hoster manipulates feedback or TI observation data to boost or reduce a SP's reputation
Countermeasure	C3	C3/C4/C8	C8	C4/C5	C1	C1	C4	C3	C2	C2	C5/C6

Tab. 5.1: Classification of Attacks on Reputation Systems – last page

ID	Description
C1	Use of (controlled) participant anonymity, e.g. using (persistent) pseudonyms
C2	Collaborative filtering based on value, frequency or hybrid cluster filtering
C3	Rigid identity management, using at least long lived pseudonyms
C4	Incentive schemes, e.g. receiving a return for truthful feedback
C5	Appropriate terms and clauses in contracts, legal regulation
C6	Enforcing confidentiality of TI data while performing computations on it, e.g. using secure multiparty computing
C7	"Pay your dues", a newcomer has to overcome intentionally set hurdles before becoming a full member of a market or community
C8	Adequate, VO application domain specific STORE system configuration, sensible SC trust preferences in query (not mandatory)

Tab. 5.2: Attack Mitigation Measures

5.3 Summary

In conclusion, the STORE reputation system architecture proves to be resilient against many attacks other reputation systems are susceptible to. Especially attacks such as A5, A6, A7, A8, A9 and A10 that exploit systems rooting their trust measure in only one uniform, highly subjective trust source, for instance feedback, are difficult to mount against STORE. STORE diversifies, uses TIs as heterogeneous, observable trust sources that are difficult to manipulate. Tampering with TI data also requires insider knowledge to shift observations consisting of sophisticated data structures towards a desired direction, e.g. reflecting less or more trustworthiness. A1, A4 and A11 are attack classes that can be mitigated with either legal or technical measures used in conjunction with the STORE reputation system, e.g. to prevent identity spoofing (A1), a suitable identity management system supports authenticated access to STORE. A11 constitutes a new class of attack assuming an untrusted reputation system hoster. It has a potential to increase in likelihood and impact with progressing globalisation and specialisation. A2 and A3 are of interest, since its design makes STORE susceptible to these attacks classes which are difficult to counteract. A2 - starting over addresses the always present problem of assigning initial reputation values to newcomers. While application domain specific settings and incentive mechanisms help to balance starting over and, on the other side, give newcomers a chance to take off in a VO environment, the bootstrap process has to continually monitored and values and mechanisms readjusted. A STORE system supporting automated decision making in more than one VO application domain offers a new variation of a freeriding attack (A3). Since this vulnerability arises to STORE's design and can only be mitigated by fine tuning the granularity with which the system recognises an organisation's trustworthiness in a VO application domain specific context, this attack is taken up again in the next Chapter 6.3.5 and evaluated in a dedicated multi-agent simulation scenario.

6. SIMULATION AND EVALUATION

This chapter presents the evaluation of the STORE reputation system within the context of different VO application scenarios. The evaluation has the purpose to fathom the hypothesis that STORE is able to provide automated decision support to a VOM for the VO member selection in all classes of VOs during their formation phase. To answer the question if this hypothesis can be accepted or is to be denied, STORE is evaluated, adopting a Multi Agent Simulation (MAS) methodology.

Section 6.1 first introduces design and implementation of the simulation framework. This entails the agent model and classes, the components and interaction sequences of the simulation framework itself and finally a concept to determine and assess an agent's Quality of Service (QoS) and production from an economic perspective across different VO scenarios. Section 6.2 contains a set of scenarios, STORE is evaluated in, complete with settings, baseline and a discussion of the results. The scenarios start with fundamental simulations, to assess STORE's basic functions such as identifying and separating different agent classes. STORE's ability to provide automated decision support to a VOM for VO member selection in all classes of VOs is assessed next. Simulations in the two VO extremes, CE and AH, are conducted. Having ascertained STORE's basic and VO scenario specific decision support capabilities, its ability to cope with dynamic changes of an agent's reputation is assessed in a sensitivity analysis. Finally, STORE's resilience against the freeriding attack (A3) from section 5.2 is assessed in a dedicated evaluation and it is compared to the Beta reputation system, the most closely related work.

6.1 The Simulation Framework

The Simulation Framework (SF) is presented top-down, starting with an architecture overview. Details about agent classes and the agent model as well as a simulation walk through including the agent matching approach are provided in the following dedicated subsections. The chapter concludes with the QoS and economic production related concept.

6.1.1 The Simulation Framework Architecture

Figure 6.1 depicts an overview of the SF architecture. The framework simulates the lifecycle of VOs, progressing in rounds. In one round, agents which represent organisations interact in a full VO lifecycle. Since the SF is designed to evaluate the STORE reputation system, both are required to interface. This takes place at two well defined interface points, first the interface providing access to TI data, second the reputation service interface delivering a reputation vector for a queried organisation based on Bayesian inference from the BN. The diagram's left, shaded section depicts the STORE reputation system as previously introduced in Section 4.1, Figure 4.1. Further details about STORE's architectural components and their relations are described there.

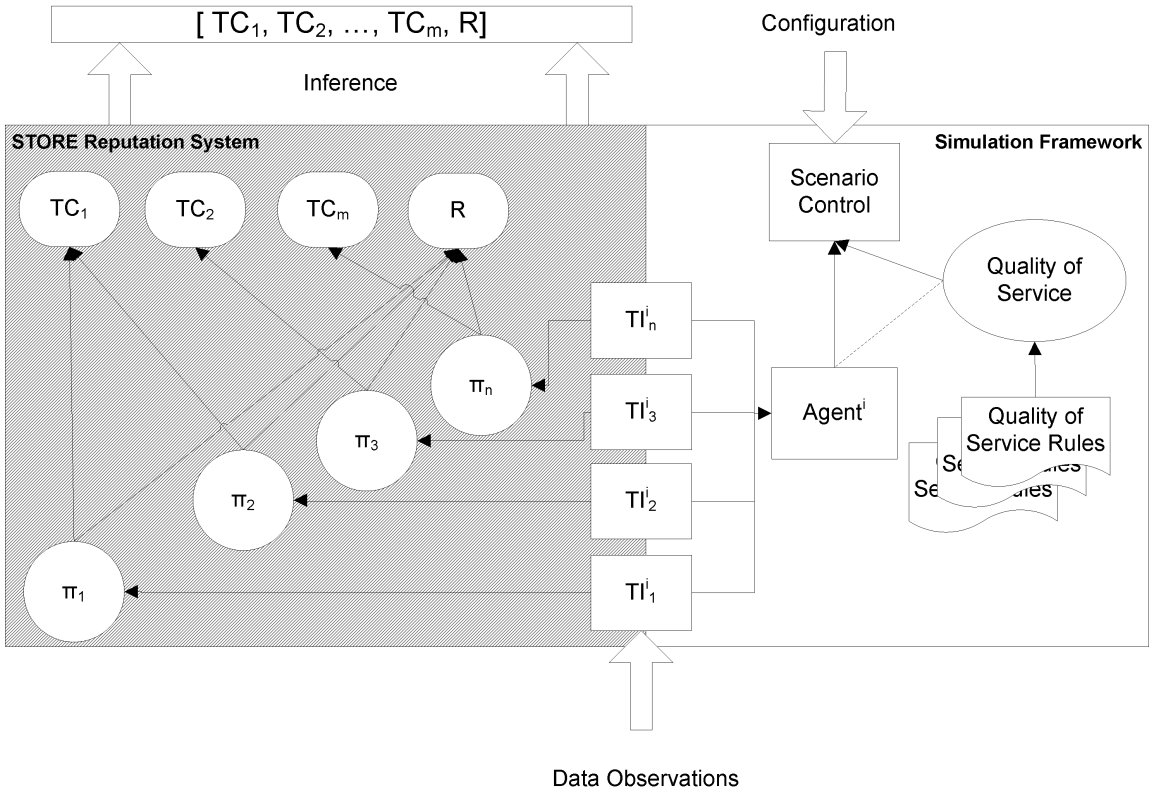

Fig. 6.1: Simulation Framework Architecture

The SF is based on the principles of a multi-agent simulation. An agent represents an organisation participating in a VO. The agent model, further detailed in Subsection 6.1.2, allows to configure agents enacting one of two possible VO roles, VOM and VO member. Agents and their behaviours are determined by their TI data. For each simulated agent $agent^i, i \in \{1, \ldots, n\}$, a dedicated, agent specific set of TI service instances in STORE's Update Layer (L1) and an equally dedicated BN instance in the Aggregation Layer (L2) is maintained. The SF generates data for each TI according to its specific update time period Δt_{upd} and injects this data immediately through the interface I1 into the respective TI service, triggering a BN update. The simulation progresses in rounds. One round represents one cycle through all VO phases from identification to dissolution phase. The details and a walk through of an individual round is provided in Subsection 6.1.3. The simulation, its agent interaction and progress, is controlled by a Scenario Control (SC) component. This component, a control class, is instantiated for each simulation based on scenario specific configuration settings. The configuration entails, among others, detailed settings about role and number of participating agents, their behaviour and VO scenario specific preferences. Further information regarding the SF's configuration options are provided in the following subsections and in a primer detailing each configuration setting in Appendix A.6. The Scenario Control component manages begin, end and the steps within a simulation round. An agent's behaviour is determined by its observed TI data, but the trustworthiness of this behaviour is differently assessed across VO scenarios. An agent representing a VO member in a long-lived

aerospace scenario may still perform reasonably when exhibiting delivery delays of up to a day. This will be intolerable for a short-lived telco scenario. The STORE reputation system takes these VO scenario dependencies into account by providing specific preference node mappings and accepting weighted reputation queries. When evaluation STORE across different VO scenarios, endogenous dimensions of freedom such as the preference node mappings must be kept fixed to achieve comparable results. In order to still be able to distinguish agent performances across VO scenarios, the notion of a QoS evaluation is introduced. When defining and configuring a simulation scenario, a set of QoS rules is defined. In each round, an agent's performance is assessed. Its performance is measured based on the agent's current TI data. This assessment puts an agent into a flexibly defined, reputation independent performance class based on his actually provided QoS.

6.1.2 The Agent Model

The SF's centrepiece is the agent model. In contrast to most related work where agents in a MAS conduct binary transactions and afterwards exchange feedback information based on simple data types, e.g. integers, the STORE model roots trust in an organisation in its set of TIs. Consequently, the agent model must entail a heterogeneous set of TIs and their stochastic model based on density functions. Those TIs, along with their respective, explicitly taken distribution assumption, represent the endogeneous variables in the VO environment that is modeled in the presented simulation framework. The TIs and their impact on an agent's reputation vector are the variables of interest in this simulation. Other exogeneous variables that may come to mind, for instance direct transaction feedback, are not a part of this modeled VO environment, though the simulation framework may be extended for that purpose.

Within the SF, an agent:

- represents an organisation.
- becomes assigned to exactly one of the two VO roles - VOM or VO member.
- becomes assigned to one of N possible agent classes, specifying its performance and behaviour.

During a simulation run, an agent of the role VOM transacts with several VO member agents, it chose to form a VO with. For the remainder of the evaluation, four agent classes are defined (N = 4):

- **Class 1**: overall good performance.
- **Class 2**: only good performance in telco scenarios.
- **Class 3**: only good performance in aerospace scenarios.
- **Class 4**: overall bad performance.

The desired amount of agents and their attribution to classes are part of the simulation setting. The agent behaviour is determined by the agent's TI data. This data is generated each TI specific update time period Δt_{upd} for each TI by drawing from an inverse of its Likelihood distribution. This approach is related to Sequential Monte Carlo[36, 122] methods. Such methods are frequently used to evaluate prior or likelihood distributions which are too complex to be evaluated analytically [109, 22]. For instance in signal theory, the goal is to evaluate a system's state that is entirely defined by those distributions. In STORE's case, the analogue is an agent's state, determined by its TI data. But this state is not exhaustively determined by the

TI distributions since those serve as evidence for the agent's BN. This is the main difference to the typical application of Sequential Monte Carlo methods. Here, the goal is to evaluate the BNs state, in particular the root nodes where the reputation measures can be inferred from.

Definition 5: A *regular TI data observation* or measurement is defined by drawing data from the TI's inverse Likelihood distribution at integer multiples of the TI's update time period Δt_{upd}.

Since agents are determined by their TI data, assigning an agent to a particular class must have an effect on the TIs. TIs are modelled with density functions, as described in Section 3.1.2 which are determined by their parameters. These parameters directly determine the corresponding distribution's expectation value which provides a qualitative measure of an agent's performance in this TI's domain. Briefly summarised, assigning an agent to a particular class determines the parameter settings of the inverse distributions which generate the TI data for this particular agent throughout the simulation run.

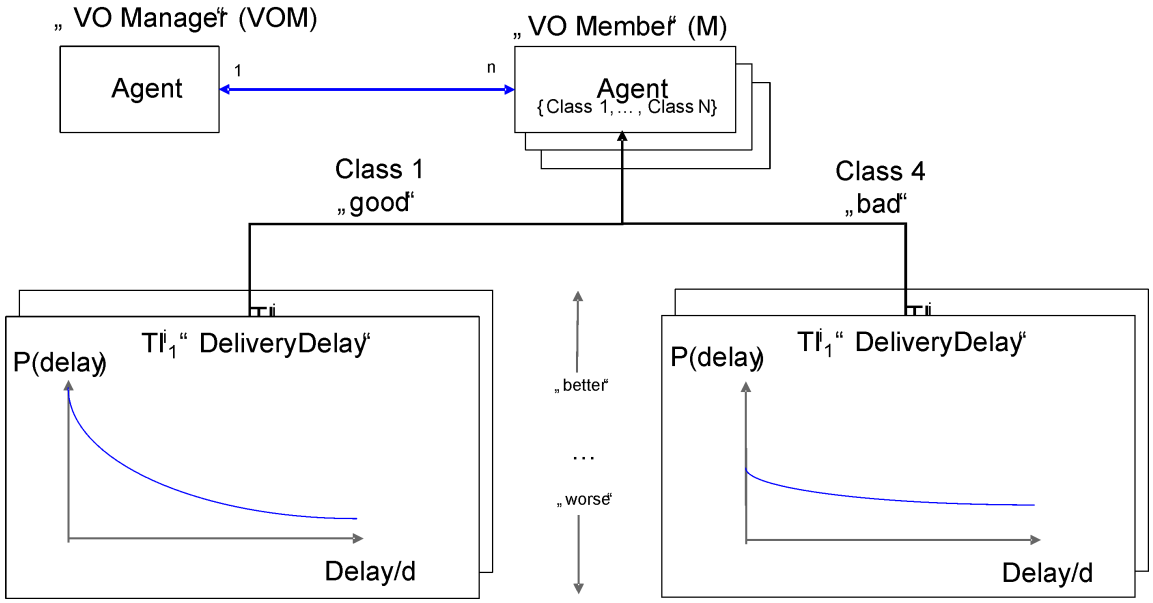

Fig. 6.2: Agent Model

Figure 6.2 illustrates the above said in an example with one TI, Delivery Delay. The figure makes the cardinalities explicit, one agent playing the role of a VOM transacts with several (n) VO member agents. Focusing on a VO member agent $Agent^i$, it is assigned to one of the N agent classes. For simplicity's sake, the example then only distinguishes between Class 1, the good performer, and Class 4, the bad performer, in the $Agent^i$'s single TI_1^i DD. DD is plotted for both classes below, denoting the delay in days on the x-axis and the probability on the y-axis. This TI is exponentially distributed with one parameter λ which determines the intersection of the plots below at zero delay with the y-axis. If $Agent^i$ is assigned to Class 1, its good performance becomes apparent in few to no observed delivery delays. In terms of the distribution,

a larger [1] λ shifts the y-axis intersection to larger positive values (left case). DD data drawn from such a parametrised inverse Exponential distribution then corresponds with a higher probability for low delivery delay. Just the opposite, if $Agent^i$ is assigned to Class 4, that results in a smaller λ shifting the y-axis intersection to smaller positive values (right case). This results in a fatter distribution tail, corresponding to a higher probability for larger delivery delays being drawn from the inverse distribution.

TI	Trust Class	Class 1	Class 2	Class 3	Class 4
Country Bond Spread	Environmental	6, 1 (6)	10, 2 (10)	6, 1 (6)	10, 2 (10)
Cash Flow Margin	Financial	3, 0.2 (3)	2, 0.2 (2)	3, 0.2 (3)	0.2, 0.2 (0.2)
Complaint Rate	Operational	2, 0.5 (4)	2, 0.5 (4)	3, 0.5 (6)	3, 0.5 (6)
Delivery Delay	Operational	2 (0.5)	2 (0.5)	2 (0.5)	1 (1)
System Downtime	Operational	3, 1 (3)	3, 1 (3)	5, 2 (2.5)	5, 2 (2.5)
Employee Fluctuations	Organizational	5, 0.2 (5)	5, 0.2 (5)	1, 0.5 (1)	1, 0.5 (1)

Tab. 6.1: Trust Indicator distribution parameters for different classes of agents

For STORE's evaluation, the SF implements[2] the six TIs presented in Section 3.1.4. Table 6.1 summarises these TIs and their TCs again and puts them in relation to the four agent classes. What was explained in the DD example above, how an agent's assignment to a class affects its TI's distribution parameters, is now expressed in concrete numbers. The table enumerates the distribution parameters for each implemented TI and possible agent class. The parameters are given in the same order as defined in the distribution summary in Appendix A.1. The number in brackets behind each parameter set denotes the resulting expectation value.
For example, when examining an agent of Class 2 (good performance in telco scenarios), the Exponential distribution's A.1.4 only parameter λ, belonging to its DD TI, is set to 2. The resulting expectation value is 0.5, meaning that this agent will probably deliver with half a day delay. The parametrisation is derived from analysed VO case studies published in [32], described with the necessary level of detail for this thesis in Section 2.3.1.

TIs are aggregated by STORE's BN towards a reputation vector. A STORE system instance maintains one BN for each agent in the SF. Comparability of an agent's TIs is ensured by the preference mapping implemented in the BN's preference nodes. The heterogeneous states of each TI are mapped onto a normalised ordinal scale of trust levels. For this evaluation the five trust levels previously introduced in Equation (3.3) are employed. Since the STORE reputation system is to be evaluated in different VO application scenarios with respect to its reputation based decision support capabilities, the STORE system configuration is kept unchanged, where possible, to allow better comparisons of simulation results from the different VO scenarios. This also includes the chosen trust preference mapping, one mapping is kept fixed for all simulation runs. The mapping is presented in Table 3.1, bottom row.

Given the TI parametrisation for the agent classes from Table 6.1 and the chosen trust preference mapping, the following Table 6.2 summarises the expected trust levels[3] for each agent class. The trust levels are determined by evaluating into which trust level the respective TI State entailing the TI distribution's expectation value is mapped into, based on the trust preference mapping.

[1] see Appendix A.1.4, the expectation value is defined as $E(X) = \frac{1}{\lambda}$.
[2] here, CF is modelled with a normal distribution since for the simulation, a standard deviation can be computed.
[3] VL - Very Low trustworthiness, L - Low trustworthiness, N - Neutral trustworthiness, H - High trustworthiness, VH - Very High

TI	Trust Class	Class 1	Class 2	Class 3	Class 4
Country Bond Spread	Environmental	H	N	H	N
Cash Flow Margin	Financial	VL	VL	VL	VL
Complaint Rate	Operational	N	N	N	N
Delivery Delay	Operational	H	H	H	H
System Downtime	Operational	N	N	N	N
Employee Fluctuations	Organizational	H	H	N	N

Tab. 6.2: Trust Indicator distribution parameters for different classes of agents

It becomes apparent that with the given choice of trust preference mapping and distribution parametrisation, the different agent classes are not expected to show big differences in their trustworthy behaviour. In fact, agents of all classes are even expected to perform within the same trust level in the TIs CR, DD and SD. They only differ in their expectation by at most one trust level in the other TIs. These narrow margins are set on purpose and constitute a "worst case" scenario when configuring a STORE system instance. It resembles an uninformed choice, when the system's owner can not rely on a tailored configuration template from a previous system instance. In conclusion, this configuration makes STORE's task more difficult to distinguish between trustworthy and untrustworthy agents and providing meaningful reputation based decision support to a VOM.

When configuring a simulation scenario, a VO application domain, e.g. aerospace or telco, is set. Therefore agents, especially the ones enacting the role of a VOM, know to which class of VO they belong to. When querying a VO member agent's reputation, a VOM sets the query's weight vector $\vec{\omega}$ according to its VO class. This vector was introduced in Section 4.1. According to Equation (4.1), $\vec{\omega}$ expresses a VOM's trust preferences for the TCs and the generalised reputation value R. Table 6.3 summarises the textual analysis from Section 3.1.4, which of the implemented six TIs carries relevance for which agent class.

TI	Trust Class	Class 1	Class 2	Class 3	Class 4
Country Bond Spread	Environmental	+	-	+	-
Cash Flow Margin	Financial	+	-	+	-
Complaint Rate	Operational	+	+	-	-
Delivery Delay	Operational	+	+	-	-
System Downtime	Operational	+	+	-	-
Employee Fluctuations	Organizational	+	-	+	-

Tab. 6.3: Trust Indicators and configuration settings for different classes of agents (+ relevant, - less relevant)

A "+" denotes the TI's relevance for this agent class, a "-" means it is less relevant. Class 1 and 4 expresses a general relevance and the opposite respectively for the TIs. Well and less well performing agents point these qualities out in all TIs. Class 2, agents specialised in telco scenarios, are selected for their increased trustworthiness in the Operational TC. Due to a telco scenario's short lifetime, mainly operational trustworthiness matters. Agents of Class 3 are especially sought for their trustworthiness in the Financial and Environmental TCs. These TCs carry more relevance with the increased life expectancy of the aerospace scenarios.

trustworthiness.

Table 6.4 expresses this TI relevance analysis now in numbers. The trust preferences of each agent class used in the simulation runs are provided. A VOM belonging to either the VO class telco or aerospace inserts the values of the respective columns as trust preferences into a query to the STORE reputation system when inquiring about a VO member agent's reputation.

Trust Class	Class 1	Class 2	Class 3	Class 4
Financial	n/a	0	0.5	n/a
Organizational	n/a	0	0	n/a
Operational	n/a	1	0	n/a
Environmental	n/a	0	0.5	n/a
R	n/a	1	1	n/a

Tab. 6.4: Agent trust preferences per Trust Class

A VOM always seeks to exploit an emerging business opportunity which leads to the formation of a VO. In this thesis, STORE's reputation based decision support focusses on supporting the VOM's decision making processes. With this setting, VOMs are always embedded in their VO application context and act therein. Therefore, the VOM is always of Class 2 or 3. Only VO members may belong to one of all the classes. Since a VOM can not fall into the Classes 1 or 4, they can not express special trust preferences for these classes which is denoted with "n/a" in the corresponding column.

The VO application domain specific trust preferences for agents of classes 2 and 3 translate into scenario specific vectors:

- for telco scenarios: $\vec{\omega}_{AH}^{T} = (0, 0, 1, 0, 1)$
- for aerospace scenarios: $\vec{\omega}_{CE}^{T} = (0.5, 0.5, 0, 0, 1)$

This choice of weights resembles a VOM from the respective application domain who is sure about which TCs are important for his VO scenario, but can not qualify the other TCs. To compensate for the latter, the generalised reputation value R is qualified with a weight of one, hereby tackling the uncertainty regarding the unqualified TCs. R aggregates all TIs equally towards a generalised reputation measure.

In summary, the agent model groups agents into classes. For the evaluation of the STORE reputation system, four agent classes are defined. The class assignment determines the agent's behaviour which in turn defines the TI's distribution parameters. Revisiting the properties of agent from Subsection 2.1.3, the agent model suits a MAS. Agents can enact exactly one of two VO roles, VOM or VO member. They interact with their environment, in particular the STORE reputation system and transact among each other as described in the following subsection. VOMs express VO application domain specific trust preferences in form of weights on TCs. The VOM communicates with STORE to obtain the member's reputation vectors and reacts on those autonomously by selecting the most reputable member agents. Furthermore, the agents, VOM and members, communicate to transact in the VO's operational phase and act rationally in the bounds of their Quality of Service ruleset which will be described in Subsection 6.1.5.

6.1.3 The Simulation Step-By-Step

The SF divides simulation runs into rounds. One round represents a cycle through all VO phases from identification to dissolution phase. In one round, a VOM agent identifies potential VO members from a set

of VO member agents, selects the most trustworthy ones from this set based on their reputation vectors and they form a VO. This VO becomes operational, the agents transact and receive in the dissolution phase a pay-off based on their QoS which measures their performance. Due to the varying lifetimes of VOs in real life, simulated rounds also represent different time intervals in real life, depending on the simulated VO scenario. For instance, in an aerospace scenario one round represents years, in a telco scenario at most days. A round therefore represents VO scenario dependent multiples of an agent's TI update time period Δt_{upd} which results in a scenario determined amount of TI updates per round. Figure 6.3 shows an overview of the individual steps that take place in the SF during one round.

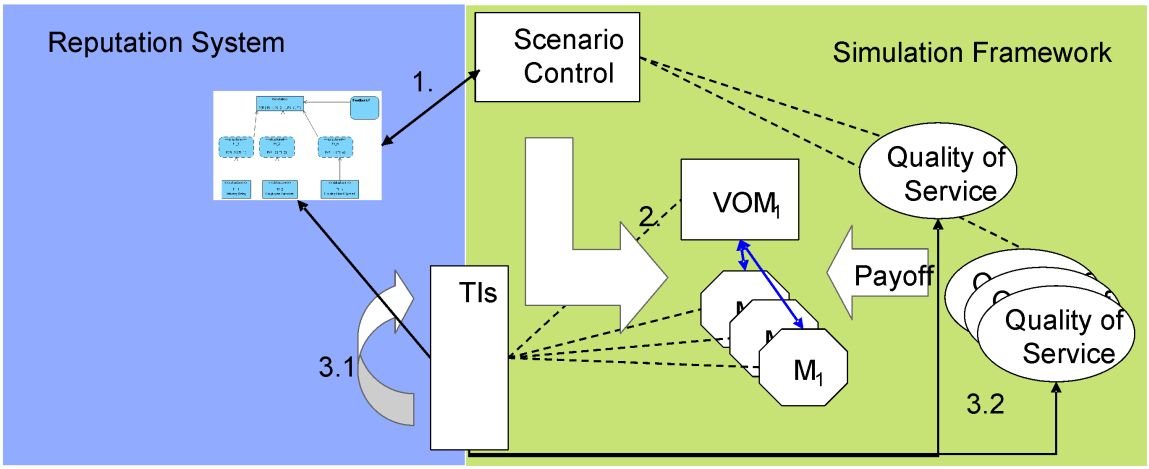

Fig. 6.3: Simulation steps in one round

The numbered sequence of steps proceeds as follows:

1. SC queries STORE for each agent's reputation vector, VOM and VO members alike (Identification Phase).
2. SC matches VOM agents with VO members based on their reputation vectors. The matched agents form a VO (Formation Phase).
3. The agents transact (Operational Phase).
    1. Each agent's TIs are updated with freshly generated data depending on the agent's class and the round duration.
    2. SC computes for each agent its actually achieved QoS for this round and the resulting pay-offs (Dissolution Phase)

The SC component receives the number of agents and their role assignment through its configuration. When the SF initialises, the agents along with their TI instances and one BN instance per agent are instantiated. Therefore, even in the first round, all necessary SF components are instantiated to perform the reputation query in Step 1. The SF offers the mechanisms to address STORE's bootstrapping problem, specifically the BN bootstrapping, from Section 4.2 in two ways. The first possibility allows to start a simulation run

from a BN that already incorporates previously learned evidence, e.g. historic TI data from previous use of this BN instance in a same or similar VO application scenario. This possibility has the advantage that a business expert can ensure a tailored, smooth system start, which may be the preferred option for STORE's productive use. In a SF for its evaluation however, STORE's usefulness across VO application scenarios is to be evaluated. It is difficult to qualitatively compare pre-learned BNs from different VO application domains, e.g. aerospace and telco, to assess if the corresponding simulation runs departed from equal start conditions. The second possibility is better suited for a MAS evaluation and employed in this thesis's simulation runs. The SF can be configured with a specified amount of so-called blind rounds. The BN is instantiated with uniform distributions in information and TC/reputation nodes. SC then starts the simulation and executes the amount of blind rounds, which are regular rounds, but their results are not attributed to the data used to evaluate the simulation run. Following that approach, all simulations depart from the same start conditions. Due to the fact that TI data is drawn from inverse distributions, simulation runs exhibit statistical differences that are addressed by repeating each run a configurable amount of times, e.g. 20 times for most VO application scenario simulations, and average over the obtained results.

For Step 2, the configuration also entails the number of VO member agents that is required to be matched to one VOM for a particular VO application scenario. Each agent computes a preference relation over the agents of the opposite role. Based on each agent's preference list, the SC component can compute the global matching. Since the STORE reputation system's decision support in different VO application scenarios is to be evaluated, the preference relation is solely based on the agent's reputation vectors. The matching algorithm used for this work is explained in Section 6.1.4. Serving as a simulation benchmark, the matching algorithm can also be configured to perform a random matching which represents a VOM's uninformed choice of VO members.

Step 3.1 leads to a full BN update for each agent upon the injection of fresh TI data. Since one round may represent varying real time durations depending on the configured VO application scenario, the amount of required TI data may vary as well.

In the SF's configuration, agents are assigned to classes that determine their expected performance/QoS. In real life, an honest organisation aims to perform with the best possible QoS, but may not achieve this goal due to unexpected circumstances, e.g. infrastructure breakdowns or even natural catastrophes. The SF simulates such behaviour by drawing an agent's TI data from their inverse Likelihood distributions. Since these data will statistically accumulate as the distribution's expectation value, the agent exhibits a behaviour as expected from its class assignment but also shows statistical behavioural variances as in real life. These statistical deviations are captured by a set of well defined QoS rules that evaluate an agent's actual ex post performance in each round. Agents carry a "virtual" bank account. The simulation framework allows to set an initial cash endowment of a virtual currency for each agent in the configuration. There, it is also possible to configure transaction costs per round that only matched agents have to pay when transacting in a VO. Agents that transacted in a VO receive in the end of each round a pay-off of the virtual currency that depends on this agent's actually achieved QoS within this round minus transaction cost. The mechanisms employed in Step 3.2 are explained in detail in Section 6.1.5.

6.1.4 Agent Matching

In real life, when a VOM recognises a business opportunity and decides to start a VO addressing this opportunity, he selects a subset of VO members from a larger set of suitable organisations during the VO's

formation phase. In the SF's virtual representation of the real world, the corresponding process is called agent matching. A VOM agent is matched with a subset of the full set of configured VO member agents. The matching is computed by a matching algorithm, that meets the following requirements:

- **Matching between two groups** - agents are divided into two groups based on their assigned VO roles, VOM and VO member agents; agents can only be matched with agents of the other role. One VO member agent is always only matched with one VOM agent.
- **Asymmetric matching** - one VOM manages a VO that consists of several VO members; one VOM agent must be matched to n VO member agents where n depends on the VO application scenario.
- **Stable matching** - when joining a VO, VOM and members agree on the collaboration's terms in a contractual agreement, their goal is to achieve the VO's objective. They have no incentive to willingly leave the VO. In the SF, the matching algorithm must achieve a stable matching that no agent has an incentive to leave the matching based on their individual preferences.
- **Reputation based matching** - the SF is designed to evaluate the STORE reputation system and its reputation based decision support for VOMs during a VO's formation phase. The matching algorithm must be able to compute a matching based on STORE's reputation vector $\vec{R}$.

The design of the SF's matching algorithm is not part of this thesis's contribution, but nevertheless at least one such algorithm needs to be chosen for the SF to perform its desired function. By its modular, service oriented design, the matching algorithms are exchangeable. To further evaluate STORE in the context of this thesis, a modified version of the well established Gale & Shapeley[43] matching algorithm is chosen. This algorithm meets all of the above requirements with the exception of the second. It was designed to solve the "Stable Marriage Problem" (SMP)[61] which introduces a model for partner selection in marriage. Men try to find the best possible partner - based on their preferences - with the goal of forming a stable relationship[4]. The SMP is solved with a symmetric matching between two groups, one man with one women, while in the case at hand an asymmetric matching between one VOM and n VO members is required. The requirement can still be met by slightly modifying the Gale & Shapeley algorithm as described in the remainder of this subsection. The algorithm produces a stable matching as proven in [43]. The following term definitions will be used in the remainder of this thesis:

Definition 6: A pair of agents from different groups, e.g. a VOM and a VO member, is called a pair of compatible agents since they can be matched to each other.

Definition 7: A combination of agents from both groups, VOMs and VO members, is defined as a **stable matching** if there is no pair of compatible agents, that prefers to be matched to each other instead of their current partners.

Each agent carries an individual preference list that orders the agents of the opposite group according to a preference relation. A VOM, for instance, carries a preference list entailing all VO member agents. To meet the requirement of reputation based matching, the preference relation is defined by imposing an agent order based on their reputation. More specifically, the order is based on the scalar product of $\vec{\omega}\vec{R}$ [5].

[4] Other examples centred around choosing students for internships in medical colleges.

[5] $\vec{\omega}$ is the VO application domain specific trust preference vector, $\vec{R}$ an agent's reputation vector of the opposite group, the result of the scalar product of both vectors is a scalar.

The Gale & Shapeley matching algorithm as employed for STORE's evaluation in VO application scenarios is described by the following pseudo code:

1. Define an arbitrary order on the group of VOMs;
2. *FOR (all VOM agents) a.maxmatch := 0;*
3. REPEAT (until each VOM is matched with a VO member) OR (each VO member is matched with a VOM)
   - A := A is (first unmatched VOM) *OR (A.maxmatch <= n)*;
   - B := A's preferred VO member;
   - IF (B is not matched)
     - Match(A, B);
   - ELSE // B is matched to A'
     - IF (B prefers to be matched to A, instead of A')
       - Free(A'); [6]
       - Match (A, B);

The strings in italic denote the adaptations made to the original Gale & Shapeley algorithm to meet the requirement of an asymmetric matching:

- The original algorithm tested the two groups for equal size, this is omitted here.
- Depending on the VO application scenario, one VOM agent is matched with the configurable number of n VO member agents; in consequence, not only unmatched, but also VOM agents not having reached their maximum of n matched VO members are operated on within the REPEAT loop.
- the *maxmatch* variable is initialised with zero.

These adaptations do not worsen the original algorithm's properties:

Assuming the matching is unstable, let A_i be matched to B_i and A_j be matched to B_j. Suppose A_i prefers B_j to B_i and B_j prefers A_i to A_j. In this case, A_i and B_j form a "blocking pair". This means that the matching process must have progressed as follows:

A_i searches for its preferred matching partner according to its preference list and becomes matched to B_i. Because matchings are never assigned with a less preferred partner, the assignment to B_i can have two reasons:

- A_i can be matched with another VO member which it prefers to B_j. This contradicts the assumption above, that B_i can not be matched with A_i.
- B_j is already matched with a VOM it prefers to A_i. Because only new matching with more preferred partners will happen afterwards, it will never be matched to A_j within the algorithm's runtime.

[6] Note that an already matched VOM becomes unmatched again.

As both possibilities lead to contradictions the generated matching is stable.

The modified version still terminates. Given an agent population of p VOMs and q ($q > n$) VO members, a matched VO member always stays matched, it may only change its partner to one higher in its preference list. The list has p entries, therefore such a change may occur $p - 1$ times in the worst case. Then the algorithm is terminated.
Looking at its performance, the worst case complexity changes from initial linear complexity with the amount of agents to quadratic for the modified algorithm. In the worst case, a stable matching is only reached after $p \cdot q$ steps.

This generated matching is "manager optimal" - meaning that under all possible stable matchings this one is the one that assigns the preferred VO members to every manager. For the evaluation of the STORE reputation system, the modified Gale & Shapley algorithm is a realistic choice. The fact that every VOM is matched to his preferred partners closely models a VO scenario in which the VOM chooses the VO members based on the provided reputation based decision support and not vice versa. Figure 6.4 depicts an example of the matching process.

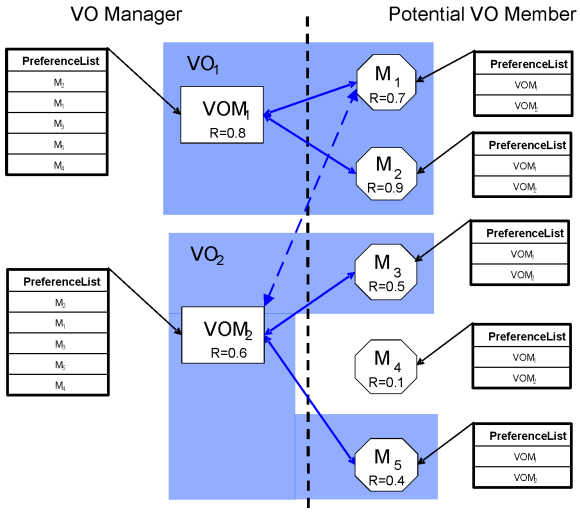

Fig. 6.4: Agent Matching

The two groups of agents are separated by the dashed line in the middle, VOMs to the left, VO members, abbreviated with a M, to the right. In the example the agent population is configured with $p = 2$ VOMs and $q = 5$ Ms, imposing an arbitrary order upon the agents. To keep the example simple, the preference lists of the agents are only based on the generalised reputation value R (which already is a scalar value), instead of the scalar product $\vec{\omega}\vec{R}$. Each agent's reputation value is printed beneath its name, e.g. VOM_1 carries a reputation value of 0.8. In the example, $n := 2$, meaning that one VOM is matched with two VO members. Each agent's preference list is shown as a table next to it, with the most preferred partner of the opposite group at its top. When executing the modified Gale & Shapeley algorithm, the following steps occur;

1. Starting with the first VOM VOM_1, it is matched with its most preferred VO member M_2.
2. The next unmatched VOM, VOM_2, is matched with its second preferred M_1 since M_2 is already matched.
3. VOM_1, still requiring a second VO member for its VO, is then matched with M_1, its second choice. This matching takes place since M_1 prefers VOM_1 to VOM_2 in its preference list, due to VOM_1's higher reputation value. This leaves VOM_2 unmatched again.
4. VOM_2 is then matched with M_3, next in its preference list.
5. While VOM_1 already reached a stable matching, VOM_2 is finally matched with M_5, its next preferred choice.

In the end, the algorithm terminates and a stable matching with two VOs is reached. VO_1 is formed by VOM_1, M_1, M_2 and VO_2 by VOM_2, M_3, M_5. Since the matching is manager optimal, M_4 remains unmatched.

6.1.5 An Agent's Quality of Service

In the real world, organisations participating in a VO, collaborate with each other by enacting an assigned business role. A seat manufacturer in an aerospace scenario with the goal of upgrading a passenger plane manufactures and ships seats to the integrator having the task of assembling the upgraded plane from its parts. A content provider in a telco scenario deliver digital contents such as digital media, weather forecasts, position information and the like. Different VO members of varying trustworthiness, working on the same task, are supposed to also exhibit variations in performance. A less trustworthy seat manufacturer delivers the seats e.g. with a higher delay. Such delay may occur due to several reasons. The seat manufacturer may be lacking personnel, may have misplanned logistics or the infrastructure suffers of natural disasters such as an earthquake. With the regular observations of such a VO member's TIs, STORE captures changes of the member's trustworthiness over time. The preference node mapping puts each TI in perspective, by mapping each TI's states to the normalised scale representing the degrees of trust.

On the one hand, this mapping is specific for a STORE reputation system instance and represents e.g. the subjective perception of trust in a VO application domain, e.g. Collaborative Engineering in the commercial aerospace industry. On the other hand, the STORE reputation system aims at providing automated reputation based decision support for the entire range of VO classes. Each VO class would therefore be supported by a STORE system instance with an application domain specific trust preference mapping. When evaluating STORE's usefulness as a reputation based decision support system in the simulation model, the virtual world, such different system configurations raise problems when comparing results across simulated VO scenarios. Suppose a seat manufacturer who manufactures seats not only for planes, but also for cars, may participate in more than one VO, e.g. an aerospace scenario in the aerospace and automotive industry. An aerospace VO tends to last longer then one from the automotive domain, hence trustworthiness in the class of organisational TIs is emphasised. As a result, the trust preference mapping of these TIs would be different. When evaluating for instance the question of STORE being able to support decisions of selecting the most trustworthy agent for a specific application domain, a simulation setting for each of the VOs from the application domains is set up, the simulation is run and results are evaluated. If the results stem from simulation runs with STORE instances based on different configurations, here, different trust preference

mappings, the simulation runs can not be outrightly compared.

In order to facilitate such comparisons, the configuration of the STORE reputation system must be kept fixed for these simulation runs that are destined to be compared. Nevertheless agents in the virtual world as do VO member organisations in the real world exhibit deviations in their trustworthy behaviour over time that requires the introduction of another measure for these deviations.

Agents, modelling real organisations, exhibit this dynamic behaviour for the following reasons:

- Even honest, real world organisations that try to deliver and perform to their best abilities sometimes suffer lapses in their trustworthy behaviour due to reasons out of their control.
- Badly performing and especially dishonest organisations exhibit great variations in their trustworthy behaviour. Many attacks described in Chapter 5 require the attacker to behave honestly at first and then radically different, depending on the class of attack that is to be mounted.
- The agent behaviour is determined by their TI data, being generated by accordingly parametrised distribution functions as described in Section 6.1.2. Data drawn from stochastic distribution functions shows variations by definition.

This additional measure, that is introduced to capture an agent's dynamic behaviour and compare agents across VO application scenarios, is called *Quality of Service* (QoS).

Definition 8: An agent's Quality of Service (QoS) is defined as a real number from the interval $[0, 1]$, $QoS \in [0, 1]$. It measures the performance or production rate within STORE's simulation environment over time.

The interval $[0, 1]$ represents the absolute scale of an agent's possible performance, 0 denoting the worst possible performance and 1 the best or optimum.

Quality of Service Rules

As previously described in Section 6.1.2, an agent's behaviour is determined by its TI data. When assigning an agent to a particular class, its behaviour is set by defined parameters for each of its TI's distribution functions. Since QoS measures an agent's performance, it is also based on the agent's TI data.

Technically, QoS measurement follows a rule based approach. Each simulation scenario is configured with a pre-defined QoS rule set (see Figure 6.1) that applies to each agent in the simulation scenario. A QoS rule set consists of one or more attributes. An attribute maps an individual TI to its fraction in the overall QoS measurement. There is no one-to-one mapping of TIs and attributes required, specific TIs can be omitted. Depending on the VO application scenario, if e.g. the TI "Employee Fluctuation" does not carry any significance for instance for an agent's performance measurement in a telco scenario, no attribute needs to be specified for this TI. Technically, an attribute defines an interval of the TI's parameter. If the TI parameters of all attributes belonging to one QoS rule fall into the respective attribute intervals, the rule evaluates to true for this simulation round. The agent's QoS is then determined by the QoS value defined in this rule.

A QoS rule set consists of an ordered list of individual rules. Rules are evaluated top-down and the first rule evaluating to true determines the agent's QoS measurement. QoS rules are evaluated for every agent

anew in each simulation round (see Figure 6.3). Rules of lower order define higher service levels, therefore, the last rule constitutes a ”catch-all” rule of the lowest possible QoS level. Empty rules, with no attributes, are allowed and always evaluate to true. A final ”catch-all” rule is frequently an empty rule.

The following Figure 6.5 summarises the previously introduced QoS measurement elements and their relations.

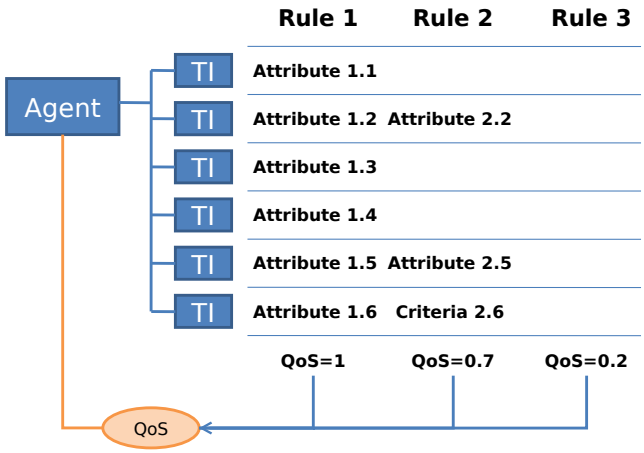

Fig. 6.5: QoS Elements and Relations

The agent carries a set of six TIs. The configured QoS rule set consists of three rules. Rule 1 contains an attribute for each TI and, in each round of the simulation, is evaluated first. If Rule 1 evaluates to true, the agent performed with the best possible QoS level of 1 in this round. In the case of Rule 1 evaluating to false, Rule 2 is evaluated next. This rule only entails attributes for three TIs and, if evaluated to true, determines a QoS of 0.7. If none of the previous rules evaluated to true, the empty Rule 3 provides a ”catch-all”, determining the comparably low QoS of 0.2 for the agent. The rule evaluation and QoS determination for the agents belongs to the first phase of Step 3.2 in Figure 6.3.

Figure 6.5 does not show the details of the attributes within the rules. Table 6.5 closes this gap with an example of a TI parameter to rule mapping.

To increase comprehensiveness, the example is based on only two TIs, Delivery Delay and Employee

TI	Rule 1	Rule 2	Rule 3
Delivery Delay [days]	$[0, 2]$	$[1, 5]$	-
Employee Fluctuations [%]	$[10, 100]$	-	-
QoS	1	0.7	0.2

Tab. 6.5: Example: QoS Attributes

Fluctuation. Assuming for instance a telco scenario, DD is by far the most important TI since quick delivery of digital content is more important than e.g. a stable organisational structure. Brain drain and other organisational effects are unlikely to influence a telco scenario that lasts for at most a day. Rule 1 evaluates first to true, if its two attributes evaluate to true. Based on the TI data generated for this round, DD's in days must be in $[0, 2]$ and the EF in % in the interval $[10, 100]$. These attributes denote the best possible performance in this example and determine a QoS value of 1. If Rule 1 evaluates to false, Rule 2 is evaluated next. When defining the QoS rule set, the decision was taken that furtheron only DD is of relevance. Therefore Rule 2 evaluates to true when its only attribute, a delivery delay in between 1 and five days, evaluates to true. In that case, the QoS is set to the value 0.7. This example also shows that non overlapping attribute intervals are not required. If all evaluates to false, the empty Rule 3 determines a "catch-all" QoS of 0.2.

VO Production

At this point, nearly all the remaining white spots and forward references from Section 6.1.3, the step-by-step description of a simulation round, are closed. Only the remainder, the second phase of step 3.2 is missing. Up to now, a single round formally started, agents were matched, hereby forming the VO, TI data for each agent was generated and the resulting QoS values were calculated. Now, in the VO's operational phase, the actual business transactions take place. Since the SF is designed to evaluate the quality of STORE's reputation based decision support, individual business transactions as conducted in a real world VO are not of importance. Instead, the SF conducts one virtual transaction in the second phase of step 3.2 (see Figure 6.3). Having transacted, the participating agents of a VO receive a pay-off in a virtual currency. To keep track of an agent's currency balance, the SF maintains a virtual "bank account" for each agent throughout a simulation run.
The agent's pay-off, as in real world VOs, depends on the agent's input into the VO. The agent's QoS value is hereby interpreted as the measure of its input, an *agent input or production rate*. Since one simulation round corresponds to a full VO lifecycle, the QoS measurement aggregates an agent's entire input for one VO. The QoS inputs of all agents participating in a VO are aggregated towards a *VO production rate* for the entire VO. Since in general, more than one VO may form in the beginning of a simulation round, the SF calculates a production rate for each VO. It has to be noted, that both, the agent and VO production rate, are rates since they are measured relative to a defined, best possible production value of 1. All rates live in the interval $[0, 1]$.

Devising a new approach of calculating the VO production rate is not in the scope of this thesis, in fact, several suitable productivity functions are available from literature to choose from [117]. Figure 6.6 summarises three suitable productivity functions graphically. As described in Section 6.1.4, the SF matches a VOM with the required number of member agents to form a VO. The transactions during the VO's operational phase are broken down to bilateral transactions being conducted between the VOM and an individual member. This approach has the advantage that it is independent of the actual VO size and can therefore cater for VOs of any size. The diagrams show the input (rate) of the VOM on the x-axis and the input (rate) of the

corresponding member on the y-axis.

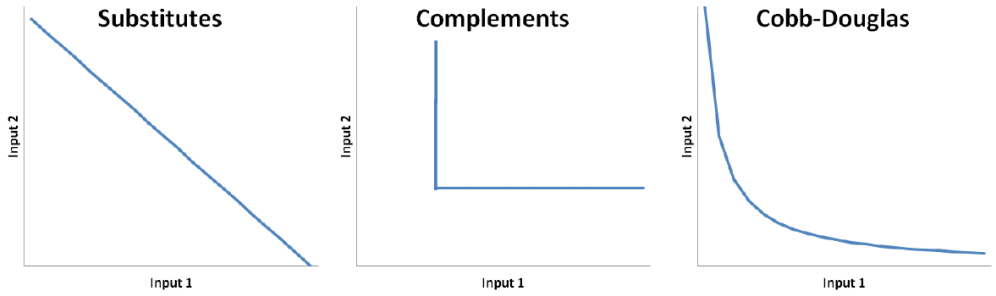

Fig. 6.6: Productivity Formulae

- The leftmost diagram can be characterised as ”perfect substitutes”. The production function sums up the available inputs. When interpreting the function in a VO context, one can observe that the VO production can still be high, when one organisation, e.g. a VO member - a supplier delivering vital goods for the VO - is performing with a low or zero input rate. This property of the function contradicts expected VO behaviour, since a VO with misperforming members is destined to fail.

- The centre diagram can be characterised as ”perfect complements”. The VO production is calculated based on the maximum of the inputs. The higher input rate is preferred in this approach, the smaller one does no longer affect the overall production rate. However, this behaviour is counter-intuitive to the expected VO behaviour. Since organisations collaborate jointly in a VO, all input rates of the involved agents should have an influence on the overall VO production. One can observe that zero input rates are no longer allowed, but in an extreme case of a very well performing VOM and a mediocre performing member, not only the VOM's performance, should dominate the VO production.

- The rightmost diagram combines the previous two approaches by multiplying both input rates. The corresponding function is called the Cobb-Douglas productivity function [23]. It is defined by $y = \prod x_i^{\alpha_i}$, $i = 2$ in the example. The possible input rates x_i are neither substitutes nor complements but exhibit similar properties as the previous two approaches: On the one hand, the Marginal Rate of Substitution (MRS) for two input factors is always positive indicating that a better quality of one member is always preferred regardless of the other members input (if it is positive). On the other hand the MRS only equals 1 if both members contribute equally much. This demonstrates that one partner's input always changes the value of the other partner's input, too. α_i can be interpreted as a dampening exponent which is set to 1 in the SF.

In summary, the Cobb-Douglas production function (with no dampening, $\forall i, \alpha_i = 1$, $i = 2$) is the best candidate for calculation the VO production, since it considers all input rates at all times.

Picking up the previous example from Table 6.5, the following table depicts the production rates for all possible transaction outcomes between a VOM and a member based on the above QoS rule set.

In each simulation round, the production rate y is calculated once for each VO member agent matched in a VO and its VOM by calculating $y = x_{VOM} \cdot x_{Member}$. The agent's QoS value servers as input factor x.

QoS manager/member	0	0.2	0.5	0.7	1
0	0	0	0	0	0
0.2	0	0.02	0.05	0.07	0.1
0.5	0	0.05	0.125	0.175	0.25
0.7	0	0.07	0.175	0.245	0.35
1	0	0.1	0.25	0.35	0.5

Tab. 6.6: Pay-off for each participating agent calculated with example QoS values

The resulting pay-off for each agent is half of the production rate, hereby equally dividing the VO's income. The pay-off for the manager of a VO is corrected for his increased number of transactions, by dividing through the overall number of transactions.
These pay-offs are always positive since the QoS values serving as the basis of their calculation are always positive. With only positive pay-offs and no costs, the agent's virtual accounts rise comparably quickly throughout a simulation run. This steep ascent of the accounts makes interpreting the simulation results more difficult, since in the real world, VOs always incur costs. Organisations participating in a VO have to maintain their (ICT) infrastructure, pay their labour force, etc. Therefore, the SF takes such costs with a fixed fee per round, so-called transaction costs, into account. Transaction costs are configured once for a simulation scenario.

Having introduced the notions of QoS, production rate and pay-off, the SF and the methodology to compare simulation results from different VO scenarios are in place. Each VO scenario simulated in a dedicated run can rely on STORE's reputation based decision support for selecting VO members while maintaining a uniform STORE system configuration. An agent's reputation vector allows to compare an agent's trustworthiness within one VO scenario, while an agent's QoS, the overall VO production rate and an agent's individual cash or pay-off rate delivers the basis to compare the agent's performance, which is based on its trustworthiness[7], across VO scenarios.

6.1.6 Implementation

The SF evaluates the STORE reputation system and, hence, requires an implementation of the STORE architecture. [50] introduces the first STORE research prototype implementing the architecture following the SOA paradigm. The prototype is implemented as Java Web Services using Apache Axis[8] as a SOAP Engine and deploying into an Apache Tomcat[9] web container. A detailed description and primer can be found in A.4. In a real application scenario, requests are supposed to originate from remote service requesters. Therefore, a web client was developed for making the reputation service operations visible to a human user. A screenshot of the web client interface is shown in Appendix A.3.

While the core SP implementations could still be used in the context of an agent based simulation, already the very first trials revealed, that Web Service (SOAP) communication imposes too much overhead for a successful simulation. In particular, the excessive memory consumption due to the required XML parsing and the resulting speed impediments discouraged the adoption of the SOA paradigm for the SF implementation.

[7] or else it would not have been selected for a VO.
[8] `http://ws.apache.org/axis`
[9] `http://tomcat.apache.org`

Therefore, only the core SP implementation of the first prototype is re-used and refactored in the SF implementation. The remaining SF components on white background in Figure 6.1 are implemented from scratch as Java objects. In all implementations, the Netica API by Norsys[10] is utilised for creating and maintaining all BNs. The API implementation is available as a library for several programming languages, the Java Native Interface version is used here.
Further details, especially all configuration settings, are provided in Appendix A.6.

This section introduced the SF and its components, implementing the agent based simulation approach to evaluate the STORE system. In particular the agent model and the QoS based measures to allow agent performance comparisons across VO scenarios and simulations were presented. The section concluded with a brief outline of the implementation.
The following Section 6.2 continues with the application of the SF and defines concrete simulation scenarios where STORE is evaluated in.

6.2 Simulation Scenarios and Setting

Having introduced the SF to evaluate the STORE reputation system with in the previous Section 6.1, this section will conduct the actual evaluation by defining, executing and analysing different simulation scenarios.

The hypothesis in question, that **STORE is able to provide automated reputation based decision support to a VOM for member selection during the VO's formation phase**, motivates STORE's evaluation. This support extends to all classes of VOs. The analysis results from this section's simulation scenarios help to either accept or deny the hypothesis.

To do so, several simulation scenarios must be defined and executed to assess STORE's desired properties that answer the research questions of this thesis:

(i) As a functional baseline, STORE must be able to distinguish between the four agent classes. A VOM must be able to select the most trustworthy agents based on their reputation vectors as computed by STORE. In a first, very basic simulation setting, the agents perform with a constant QoS.

(ii) The next two simulation scenarios introduce a VO application context. First, STORE's reputation based decision support is analysed in the context of an aerospace then telco scenario. In these simulation scenarios, VOMs express their preferences with the VO specific weight vectors.

(iii) An important advantage of STORE is the ability to take dynamic changes in an VO member's trustworthiness into account. The second simulation scenario conducts a sensitivity analysis with agents that change their QoS dynamically.

(iv) Having introduced VO application scenarios, the freeriding attack A3 from the previous Chapter 5 is revisited. In a dedicated simulation scenario with a VO context, STORE's ability to distinguish between trustworthy and freeriding agents that appear to be specialised in another VO application domain, is analysed.

(v) Finally, STORE is compared against another reputation system, the Beta reputation system that was described as related work in Section 2.1.

[10] Netica and the Netica API are available as a free limited version at `http://www.norsys.com`

Before the simulation results can be presented, the simulation configuration and setting used for these scenarios are defined.

Since the SF is designed to evaluate STORE in arbitrary VO application scenarios, a large number of configuration parameters is required to flexibly capture the complexity of VO environments in a simulation setting. Many of these parameters are used for debugging purposes of the SF itself, these are not of interest in this section. Here, only the parameters that directly influence a VO scenarios configuration are presented. The entire set of parameters along with a primer for their application is provided in Appendix A.6.

The following Table 6.7 summarises the scenario relevant parameters.

Parameter	**Aerospace Scenario**	**Telco Scenario**
# of VOM agents/class	(1/0/0/0)	(2/0/0/1)
# of M agents/class	(2/2/2/3)	(1/2/2/2)
# of M per VOM	5	2
# of rounds	20	50
# real time/round	years	days
# of blind rounds	2	2
Initial endowment	0	0
Transaction cost	0.15	0.15
# of repetitions	50	50

Tab. 6.7: Simulation parameters and settings per scenario

The table contains a column for each of the VO scenarios, aerospace and telco, STORE is evaluated in. The baseline, each of the scenarios can be compared with, is a scenario where VOs are formed based on random matching. Instead of having a VOM perform a reputation based VO member selection with the algorithm from Section 6.1.4, members are randomly selected. The scenario setting for such a random selection is not explicitly listed in this table. If a random matching benchmark is provided for a scenario, the scenario's setting applies as well for the benchmark.

A scenario setting first defines the agent population for each VO role and class. The basic proportions originate from the VO scenarios as described in Subsection 2.3.1 and in more details, the TrustCOM VO classification[32]. In an aerospace scenario, one VOM of Class 1 and five member agents of classes 1 to 3, with three members of Class 4 interact. In a telco setting, two VOMs of Class 1, one of Class 4 with one VO member agent and two of each of the Classes 2 to 4, collaborate. Longer lasting aerospace scenarios, tackling challenging engineering problems that require expert VO members from many scientific fields, require more VO members as compared to telco scenarios. Due to the shorter lifetime of telco scenarios, the setting requires less VO members to be matched into a VO, but allows for more VOs to be formed in parallel due to the larger amount of VOMs.
In an aerospace scenario, five member agents must be matched with one VOM for the successful formation of a VO, in contrast to two in a telco setting. The latter fact takes the larger amount of participants, such as digital content and infrastructure providers, in an ad-hoc service provisioning environment into account.
An aerospace scenario simulation runs for 20 round which represents several years of real time for a VO to complete its lifecycle in one round. The shorter lived telco scenario simulations run for 50 rounds which accounts for days in real time.

Each simulation departs from the same base configuration which includes an empty BN for each agent. When injecting the first generated TI data observations into the BN's information nodes, the TC's and the generalised reputation node R's distributions may exhibit oscillating behaviour over some time. The probability of observing such an initial tuning effect increases with the number of - indirectly, via preference nodes - connected information nodes. To avoid result set pollution of this oscillating effect, a number of blind rounds can be configured. The simulation then starts off executing rounds, but begins with the output of result data, the desired key figures, only after the number of blind rounds has elapsed.
For the computation of the VO's productivity, the overall and agent specific cash flow, an initial endowment can be defined for each agent. This parameter is currently not used and set to zero for each agent. To avoid the unrealistic increases of the agent's virtual accounts that were mentioned in Section 6.1.5, a fixed transaction cost per round that each member agent that is matched in a VO has to pay, can be defined. The value is set to 0.15 units of the virtual currency for each scenario.
To avoid statistical errors in entering the scenario analysis, each simulation setting is run with identical configuration for 50 times and the presented results constitute the averaged figures from the 50 runs.

6.3 Evaluation

At the very least, reputation systems are designed to distinguish the subjects, they compute reputation values for, based on a defined trust measure. Many systems, especially the ones based on a rather simple and subjective trust measure such as feedback, can only distinguish coarse grained between well and badly performing subjects. In STORE's case, the reputation system is designed to also distinguish between subjects, the agents representing organisations, that are specialised in specific VO application domains within the context of, not necessarily the same, VO application domain. This ability of a reputation system to distinguish between subjects of varying trustworthiness is referred to as the ability to separate these subjects in a simulation scenario based on their computed reputation. The following subsections evaluate STORE in the above enumerated simulation scenarios based on this fundamental ability.

6.3.1 Basic Evaluation

In this section, first, only the very basic functional aspects of STORE are evaluated. The agents of the different classes are configured by their class assignment to exhibit one type of behaviour that is not changing throughout the simulation run. The key figure of interest in this very basic scenario is the computed reputation for agents of the different classes and how well STORE separates the representatives of the four classes from each other. A specific VO application domain is not introduced at this point, VOM agents are not using scenario specific preference vectors. The telco setting from Table 6.7 is used for the simulations in this section. With no weights for the different TCs for the reputation vector $\vec{R}$, defining to a preference vector of $\vec{\omega}^T = (0, 0, 0, 0, 1)$. Only the generalised reputation value R is observed, aggregating all of the information nodes in the BN. In more detail, the value R is the expectation value of the self named node's distribution in the BN. The BN's topology and TIs can be consulted in Figure 4.3.

The simulation results are depicted below in two charts for each simulation. The left chart illustrates the key figure that is evaluated in the simulation. The key figure entails in nearly all cases a reputation measure of the agent population. This reputation measure either consists of the entire reputation vector or its components, the generalised reputation or a trust class specific measure. If necessary for the interpretation, a chart presenting the alternative, reputation independent QoS measure, is depicted to the right. QoS is interpreted

as the measure of the individual agent's input into the VO. The QoS measure is also based on the TI data, defining the agent's actual performance throughout the simulation, but is defined by the application domain specific QoS rule set as described in Subsection 6.1.5. For discussing the agent's performances in a chart, an agent of each class is picked as the class's representative. The x-Axis in each chart shows the simulation rounds, omitting the blind rounds.

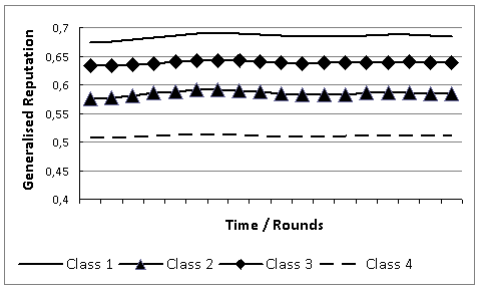

Fig. 6.7: Generalised Reputation of the four agent classes changing over time in a telco scenario

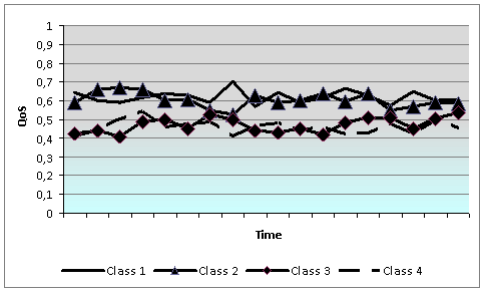

Fig. 6.8: QoS of the four agent classes changing over time in a telco scenario

Figure 6.7 and 6.8 present the results of the described simulation setting. Most importantly, the well separated contrasting agents of classes 1 and 4 stick out. The generalised reputation values in the left chart are continually very well separated, putting them in between 0.15 and 0.2 apart. Also, the two remaining agents, specialised in telco (Class 2) and aerospace (Class 3) VOs, are continually well separated, though standing only 0.07 in absolute value apart. The agent separation regarded in these comparably small absolute values is put into perspective with the right hand chart. This chart plots the agent's QoS measurements. Looking again at Table 6.1, listing the parametrisation of the agent's inverse TI distributions, and their discussion in the accompanying Subsection 6.1.2, one has to be reminded that the definition of the agent classes happened, having a worst case scenario in mind. Worst case means that the agent behaviour, determined by the parameters of their inverse TI distribution from which their data is drawn, is not far apart. The agents participating in a simulated VO exhibit, on purpose, similar delivery delays, cash flow margins, etc. to make it more difficult for STORE to separate them based on a computed reputation measure. The right hand chart now proves that the configuration of a worst case scenario succeeded. All agent's QoS measurements appear tightly packed. While the agents of class 1 and 4 show continually disparate QoS measurements, the agents of class 2 and 3 are depicted with even intersecting lines. As mentioned above, the charts show the averaged results of 50 runs of the same scenario. Still, especially visible in the right hand chart, agents follow a stochastic behaviour within the boundaries of their TI distributions.
Summarising the result of this simulation, STORE is able to continually and distinctly separate agents of all four classes even in such a worst case scenario.

It should also be noted that, assuming the same STORE system configuration, the generalised reputation value R is scenario independent. Regardless of a telco or aerospace setting from Table 6.7, the values are the same. The QoS measurements on the other hand are, due to the scenario specific QoS rule sets, strongly scenario dependent. These rule sets become important in the following Subsections 6.3.2 and 6.3.3 where they are listed in the Tables 6.8 and 6.10. Figure 6.8 shows the QoS measurements with the telco rule set. As an advanced preview and to be complete, Figure 6.9 shows the measurements of the same scenario for all

four agents with the aerospace rule set. Comparing both charts, it can be observed that the agents of Classes 1 and 4 show similar good, respective bad performance results, as expected from the class definition as well. The plots of these agents set the boundaries for positive, respective negative QoS values. Changes and more dynamic behaviour can be observed for the specialised agents in between these boundaries. They switch their ranking in the QoS measurement depending on the scenario. In the aerospace scenario, Figure 6.9, the aerospace specialised Class 3 agent shows the better QoS values compared to the telco specialised Class 2 agent. In a telco scenario, Figure 6.8, their measurements are switched. The introduced QoS measurement can therefore be considered as a well defined scenario dependent, but reputation independent measurement, able to resolve and assess an agent's performance.

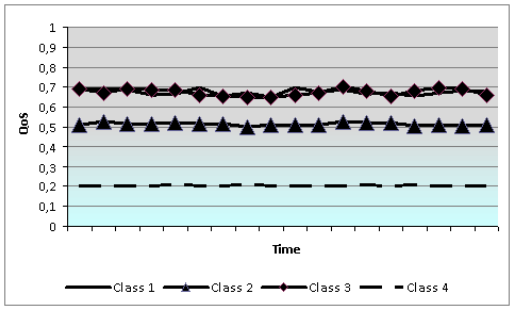

Fig. 6.9: Quality of Service (QoS) for the agents in an aerospace scenario

An additional observation about the specialised agent's reputation in Figure 6.7 should also be mentioned. On the first hand, it appears to be counter-intuitive that the class 3 agent receives a better reputation measure than the class 2 agent, while their QoS measurement is in most rounds reversed. Agents, specialising in VO application domains, appear more trustworthy in the TCs relevant for this domain (as measured by the TI observations from these TCs). But TCs are not directly comparable and since no VO context in form of a preference vector or a VO specific STORE configuration[11] was considered, such results may appear. Interpreting this result from another perspective suggests, that without preference vectors the selection of well performing, but specialised agents, e.g. of classes 2 and 3, becomes much more difficult. In this example, running with a telco setting, the wrong agent of class 3 with better generalised reputation value would have been selected. Here, telco scenario specialised agents (Class 2) excel in their operational qualities while agents in aerospace scenarios e.g. are considered trustworthy due to their financial reliability. Figures 6.10 and 6.11 take a closer look at the development of the Operational and Financial TC nodes within the same simulation scenario. Here, the y-axis depicts the expectation value of the named TC node's distribution.[12]

While the agents of the classes 1 and 4 again show their expected contrasting behaviour, this drill-down shows the source for the specialised agent's differing reputation as seen in Figure 6.7. Telco specialised agents are more trustworthy in operational aspects, which is measured by operational TI data observation. This influences the generalised reputation value R (through the preference nodes), but also manifests in the operational TC node as seen in the left hand chart. The same holds true for aerospace specialised agents and their higher trustworthiness in financial aspects in the right hand chart. Only the introduction of the TC nodes in the BN's topology makes it possible that exogenous VO contexts in the form of a VOM supplied

[11] The STORE configuration, e.g. the preference mapping, is kept fixed on purpose to allow comparability of results across VO application scenarios.

[12] The results are plotted on a scale of [0, 10] on the y-axis for increased readability.

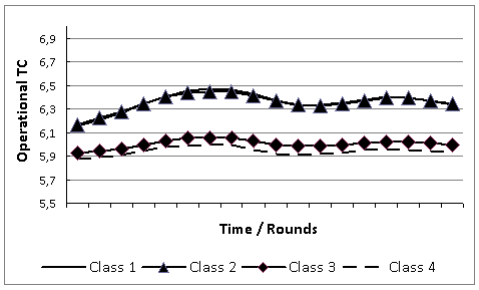

Fig. 6.10: Reputation for "Operational" TC for different agent classes

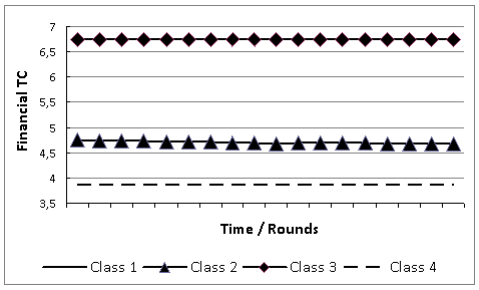

Fig. 6.11: Reputation for "Financial" TC for different agent classes

preference vector can be used for an improved partner selection during a VO's formation phase, even selecting the most trustworthy, specialised partner.

To substantiate this claim, the same scenario is run again, but using the scenario dependent preference vectors from Section 4.1, Reputation Interface. The y-axis of the charts in Figures 6.12 and 6.13 plot a scenario dependent metric defined as:

- $\psi_{CE} = \vec{R} \cdot \vec{\omega}_{CE}$ with $\vec{\omega}_{CE}^T = (0.5, 0.5, 0, 0, 1)$
- $\psi_{AH} = \vec{R} \cdot \vec{\omega}_{AH}$ with $\vec{\omega}_{AH}^T = (0, 0, 1, 0, 1)$

The metrics calculate a scalar product of a scenario dependent preference vector, as supplied by an informed VOM, and the STORE computed reputation vector.

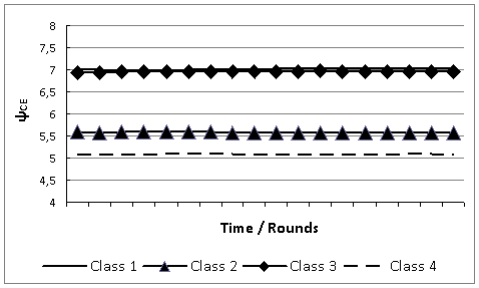

Fig. 6.12: Metrics ψ to generate the preference relation with weighting vectors ω_{CE}

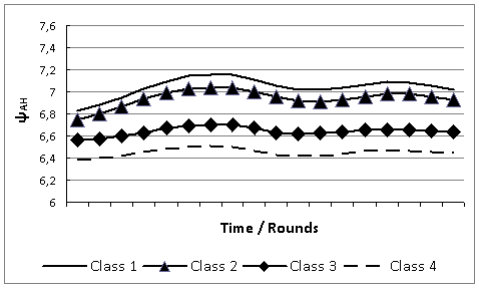

Fig. 6.13: Metrics ψ to generate the preference relation with weighting vectors ω_{AH}

Observing the plots of both metrics, the agents of classes 1 and 4 still remain with their highest, respectively lowest value. But, in the same scenario setting as in Figure 6.8, the specialised agents are plotted as expected. Taking the preference vector into account, the metric values an aerospace specialised agent (Class 3) higher than the telco specialised (Class 2) in an aerospace scenario (left hand chart). The valuation is reversed in the right hand charts in the telco scenario. This metric ψ, along with informed preference vectors ω provides decision support to a VOM to select the most trustworthy agent in a VO's formation phase. It will be used, as described in Section 6.1.4, for the SF's agent matching in the following section's scenarios.

6.3.2 Scenario Specific Evaluation - Aerospace Scenario

This subsection tackles the VO application scenario oriented evaluation. The first scenario is the aerospace one. Since such application oriented simulation settings produce results of higher relevance for STORE's evaluation, meeting its design goals, more effort is put into the results analysis. Besides interpretation and discussion, also statistical methods are now applied. Since this is the first subsection dealing with these statistical methods, it starts off with a few words on background.

Statistical tests that aim at the qualitative evaluation of e.g. experimental results can be of a great help in assessing stated research questions. However, such statistical tests are frequently wrongly applied and rather easily, false conclusions can be drawn from test results.
Statistical tests never deliver any certainty or absolute assurance regarding a stated research goal in form of a hypothesis. Test results are rather probabilities which serve as grounds to accept or deny the hypothesis to a previously defined acceptance niveau. In proper application, the hypothesis is to be stated before the experiment.
So far in STORE's evaluation, only the most basic evaluation scenarios were performed. The results from simulations with application domain specific settings, presented in this subsection, allow for a meaningful application of statistical tests. In particular, a meaningful hypothesis can be stated.

In this subsection, simulations of VO scenarios, set in the two domains of Collaborative Engineering (CE) and Ad-Hoc Service Provisioning (AH), are performed. The STORE reputation system is designed to provide automated, reputation based decision support for a VOM's partner selection process during the formation of a VO. Not a particular (class of) VO, but for all classes.

A well established candidate for testing continuous observations is the (Student) t-Test. The t-Test distinguishes group results on the basis of a parameter, e.g. an expectation value μ. It therefore belongs to the class of *parametrised tests*. The t-Test assumes that the involved random variables are normal distributed. Of interest in the simulation results are the differences of the generalised reputation value and, by applying a trust preference vector, the remaining components of the reputation vector $\vec{R}$.

All components of $\vec{R}$ are expectation values of distributions that are aggregated from identically distributed, stochastically independent - see (A.1.2) - random variables. The individual TI's posterior distributions become comparable, when the Π-node's random variables are computed at each update. The latter random variables are computed from the conditional probability with the TI's random variable, $P(\Pi_i|TI_i)$, $i = 1, \ldots, n$. Since the Π-node distributions are defined with the same amount of states, in the same domain and are identically bootstrapped, they are identically distributed. They inherit their stochastic independence from each other from the TI nodes's stochastic independence. The states cover the interval $[0, 1]$, the domain is also defined in $[0, 1]$, ensuring finite expectation values and variances for all Π distributions. As described and visualised in the diagrams of Subsection 4.2.1, the expectation value of a Π node's random variable falls into the trust level interval where the corresponding TI states amount for most data observations. The Π node's random variable values increasingly decay, the farther the trust level happens to be from the expectation value, leading to a bell shaped support. Judging from the numerous simulation scenarios, the observed generalised reputation and TC nodes, that aggregate the homogeneous Π nodes, are even closer to a normal distribution (see Appendix A.3 for an example, shown in the main window). Unfortunately, it is not possible to draw more conclusions about the reputation vector component's distributions. Especially describing it in a closed, algebraic form is not possible. While their expectation values, e.g. from Trust

Class i, TC_i, can be clearly determined as the product of the aggregated, stochastically independent j TIs, $E(TC_i) = \prod_{l=1}^{j} E(TI_l)$, their distributions as products of other distributions may be very heterogeneous. No applicable theory for this case is available, while for e.g. sums of independent, identically distributed random variables the Central Limit Theorem A.1.5 applies. [59] list several examples of similar cases from his simulation results in his dissertation, [58] provides a shorter overview. Based on the experience of numerous analysis's of SF result sets, STORE's reputation vector components constantly showed an approximate normal distribution. Also in practice, the t-Test is frequently applied to result sets, that can not be proven to be normal distributed, but show a good fit, as judged by the scientist.

Besides the already good fit of requirements, the t-Test has several other advantages because of which it is chosen as the preferred statistical test in this evaluation chapter:

- The t-Test is able to recognise differences between test groups with a high probability even when only few samples are available (power of the test).
- The t-Test maintains all of its desired properties even when the random variables are only approximately normal distributed or when false data slightly pollutes the sample base.

STORE's decision support mechanism returns a reputation vector $\vec{R}$ to a requesting VOM. The evaluation focuses therefore on the reputation vectors for different agent classes. The question to be evaluated is "Can STORE distinguish agents from the defined classes?". This question can be subdivided into a worst and average case.

The worst case assumes an uninformed VOM who expresses no TC preferences in his reputation query. Therefore, only the generalised reputation value R enter his decision making process.

The average case assumes an informed VOM who expresses well defined TC preferences. The following preference vectors, previously introduced in Section 6.1.2, is used: $\vec{\omega}_{CE}^{T} = (0.5, 0.5, 0, 0, 1)$

Re-iterating, the weights emphasise the TCs of environmental and financial TIs[13], as well as the generalised reputation value R. The more frequently observed financial TIs capture an agent's short term development in trustworthy behaviour and complement the observations of an agent's long term development with less frequently observed environmental TIs.

The resulting QoS rule set in Table 6.8 also reflects this preference in TCs.

QoS	1	0.7	0.5	0.2
Country Bond Spread	0-4	4-7	-	-
Cash Flow Margin	5-∞	4-5	$2-4$	-
Complaint Rate	-	-	-	-
Delivery Delay	-	0-3	-	-
System Downtime	-	-	-	-
Employee Fluctuations	-	-	-	-

Tab. 6.8: CE scenario QoS interval rules for TI data

[13] a weight of 0.5 as applied to maintain the same sum of weights as applied in the next section's telco scenario preference vector

The table should be read in rows, rules are ordered from left to right. "-" states, that position is not observed. Rows containing only "-" do not contribute to the QoS rule set. Four QoS intervals are defined for the aerospace scenario. The intervals are interpreted as follows:

- $[1, 0.7[$ - high quality and exceptionally well performing agent
- $[0.7, 0.5[$ - well performing agent
- $[0.5, 0.2[$ - poorly performing agent
- $[0.2, 0]$ - exceptionally poorly performing agent

The intervals are chosen that the TI parametrisation for the agent classes as enumerated in Table 6.1 are mapped taking the aerospace scenario's application domain into account. The agent classes 1 and 4, without a particular VO application domain specification, are mapped to the higher, respectively lowest QoS intervals. The mapping favours the aerospace specialised agents of class 3, mapping them to [0.7, 0.5[, while class 2 agents are borderline between this and the next lower QoS interval. It has to be stated again that these are no fixed mappings, since the agent behaviour is determined by their stochastic TI data, which is drawn from inverse Likelihood distributions. As in real life, agents aim at achieving their best performance, but sometimes an agent can do better or worse, compared to their expected behaviour.

The aerospace setting from Table 6.7 is used for the following scenarios. A collaborative engineering VO environment is characterised by fewer potential VO members compared to other VO environments, partly due to the high initial investment the partners have to take, e.g. when integrating their ICT infrastructures. Due to this fact, only two member agents are matched with one VOM agent in the simulation.

First, as in any other scenario, the separation of Class 1 and 4 member agents is paramount. As a basic function of any reputation system, it must be able to differentiate between well and baldy performing agents. The next interesting question targets the agents of Classes 2 and 3. STORE should be able to separate them by reputation as well, allowing a VO manager to select the most trustworthy members for his VO. Each of these scenarios is evaluated in the average case, assuming an informed VOM expressing the correct VO specific trust preferences, and worst case, assuming an uninformed VOM with no trust preferences at all. The hypothesis is typically formulated in a negated form:

H_{14}: STORE is not able to differentiate between agents of Classes 1 and 4.
H_{23}: STORE is not able to differentiate between agents of Classes 2 and 3.

The hypothesis's are indexed according to the scrutinised agent classes, to avoid confusion with other conventional naming schemes. The goal is to deny the hypothesis in each case.

As shown above, the setting meets the requirements of a t-Test. The one sided t-Test is applied to each of the result sets. In each average case, the metric ψ_{CE}'s time series of the involved agents over the executed rounds are tested for equality of their expectation values. ψ_{CE} takes, by definition, the trust preferences $\vec{\omega_{CE}}$ into account. For the worst case, only the time series of the generalised reputation values R are tested for equality of their expectation values, assuming a trust preference vector $\vec{\omega} = (0, 0, 0, 0, 1)$. The conventional error niveau of $\alpha = 5\%$ is applied. The t-Test then produces as its result the p-value, the probability, of accepting the hypothesis. The following Table 6.9 lists the p-values of the hypothesis under each case.

As a first, very encouraging, observation, it becomes apparent that both hypothesis, in the informed and uninformed VOM case, can be denied due to highly significant p-values well beyond even the $\alpha = 1\%$ niveau. Interpreting this test, this means that in each case agents of classes 1 and 4, as well as class 2 and 3

Hypothesis	average case	worst case
H_{14}	$2,54355 \cdot 10^{-43}$	$1,46525 \cdot 10^{-36}$
H_{23}	$1,31779 \cdot 10^{-63}$	$3,17392 \cdot 10^{-28}$

Tab. 6.9: t-Test results in the aerospace scenario

are very well separated relying on STORE's reputation based decision support.
Looking more closely at the results, the p-values for both hypothesis in the average case are lower than in the worst case. This re-confirms ex post the classification of the average and worst case by several powers of ten. When comparing the p-values of the hypothesis in the same case with each other, an interesting fact can be noted. While the rather polarised agent classes 1 and 4 are at all times well separated, this task is more difficult to achieve for the classes 2 and 3 with similarly well performing agents. This fact is graphically visible in Subsection 6.3.1's diagrams, e.g. for separation based on the generalised reputation value R or based on trust preferences with the metric ψ_{CE}, but becomes especially apparent under the reputation independent QoS measure in Figure 6.9. However in this table, H_{23}'s average case stands out. While in the worst case, H_{14} can be denied with even lower p-value than H_{23} as expected, it is the other way round for the average case. This result proves that by even applying a comparably simple metric as ψ_{CE} along with well expressed trust preferences emphasising the relevant TCs for the current VO application domain, STORE is able to even separate agents of very similar trustworthiness and performance with high significance.

Having assessed STORE's output vector with respect to its ability to provide reputation based decision support in a simulated aerospace scenario environment, the following paragraphs are now analysing STORE's impact on the same VO environment's welfare. The regarded key figures here are an agent's productivity measured by its cash flow rate. This rate is created through the received pay-off, after having successfully participated in a simulated VO. The productivity measure is based on an agent's reputation independent input into the VO, measured by its QoS value. The following three charts show the results, Figure 6.14 depicts the cash flow rates of agents that are randomly matched into VOs, the benchmark scenario. Figure 6.15 shows the results of a worst case scenario, Figure 6.16 of an average case scenario with reputation based agent matching supported by STORE. When measuring a VO member's productivity, a bad case occurs, when an uninformed VOM wrongly expresses his trust preferences, e.g. for an unrelated VO application domain. The worst case then happens, if a VOM expresses trust preferences for a diametrically opposed VO application domain which occurs, according to the relied upon VO classification, by selecting telco scenario trust preferences in an aerospace scenario. In the average case, the correct aerospace trust preferences are applied.

On a first glance, the superior cash flow rates in the average case in Figure 6.16 and even in Figure 6.15's worst case compared to the random matching benchmark can be observed. Even with an uninformed VOM, misinterpreting the trust requirements for his VO application domain, STORE supplied reputation based decision support is preferable than none at all. The cash flow rates in the benchmark are all centred around or well below zero, since even well performing agents are randomly matched with Class 4 agents and receive, as a result, reduced pay-off in this round. With STORE supported agent matching under the aerospace scenario simulation setting, VOMs can form VOs with the most trustworthy agents. There is no shortage in the number of available member agents, therefore Class 4 agents, carrying the lowest reputation rating, are not matched into VOs at all. Since they still have to pay transactions costs per round without receiving any pay-off, their cash flow rates are dropping. One can even interpret that in an uninformed VO environment, lacking reputation based decision supported such as in the random matching benchmark, encourages and

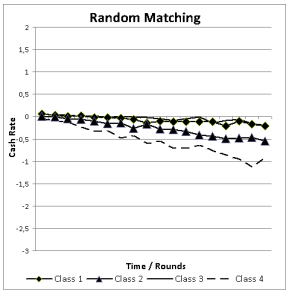

Fig. 6.14: Cash rates for the four agent classes in a random matching aerospace scenario.

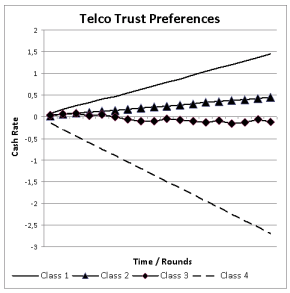

Fig. 6.15: Cash rates for the four agent classes in an aerospace scenario with wrong trust preferences.

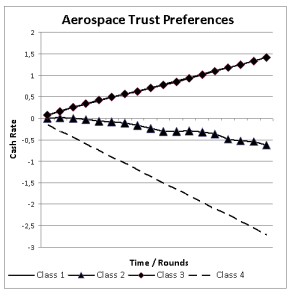

Fig. 6.16: Cash rates for the four agent classes in an aerospace scenario with correct trust preferences.

sustains bad performance. This encouragement manifests in Class 4 agents, randomly receiving a small pay-off on account of honest agents.
In both, average and worst case, the polarised agents of Class 1 and 4 set the boundaries for achievable cash rates. More interesting are the rates achieved by the specialised agents. In the right hand, average case with the correct aerospace trust preferences, agents are selected based on their reputation in the correct VO context. Aerospace specialised Class 3 agents are continually better off than telco specialised Class 2 agents, having a cash rate of the same gradient as a Class 1 agent. Also based on the simulation setting, telco specialised agents are considerably less matched into VOs, accounting for their negative trend in the cash rate. Class 3 and 2 agents are therefore well separated through STORE's reputation rating, visible in their cash rate's gradient differential.
In the worst case, the telco specific trust preferences lead to a more frequent selection of Class 2 agents on account of the aerospace specialised Class 3 agents having a nearly constant cash rate around zero. But the gradient differential between both classes is considerably smaller then in the average case. Selecting a telco specialist agent, e.g. an electronic storage service provider, for an aerospace scenario does not inadvertently lead to the VO's failure, if this member can still enact the required business role. The overall VO's profit may suffer slightly, e.g. due to increased storage service round trips or access times, but the VO is still perfectly able to master its business goal.

Having successfully analysed - with very encouraging results - how STORE's reputation based decision support to the VOM in an aerospace scenario environment, a similar analysis is now performed in the following subsection for a telco scenario.

6.3.3 Scenario Specific Evaluation - Telco Scenario

The telco scenario environment, based on the telco setting from Table 6.7, is used for the following scenarios. An Ad-Hoc Service Provisioning VO environment is characterised by a larger number of VO members compared to other VO environments. Due to the environment's agility, the speed of setting up a VO and its quick cycle through the phases, as well as the higher ICT reliance and integration, the bar is lowered for newly joining members. Due to this fact and the larger number of required electronic service and infrastructure providers, five member agents are matched with one VOM agent in the simulation.

Again, as in the aerospace scenario, an informed VOM in a telco scenario expresses his preferences in a trust preference vector: $\vec{\omega}_{AH}^{T} = (0, 0, 1, 0, 1)$
The sum of all preference vector components is the same for all scenarios, avoiding the introduction of a weight related bias when comparing simulation results across VO scenarios:

$\sum_{i=1}^{5} \omega_{i_{AH}} = \sum_{i=1}^{5} \omega_{i_{CE}}$

The weights emphasise the TC of operational TIs and the generalised reputation value R. This choice reflects the properties of ad-hoc service provisioning VO environments with many business transactions - and therefore frequently formed new VO structures - per day - but of a low financial volume. The rapid development requires an organisation to achieve excellence in operational properties for being rated as a trustworthy VO partner.

This gives rise to the definition of the following QoS rule set for telco scenarios in Table 6.10.

QoS	1	0.7	0.5	0.2
Country Bond Spread	-	-	-	-
Cash Flow Margin	-	-	-	-
Complaint Rate	0-2	0-4	0-6	-
Delivery Delay	0-1	0-3	-	-
System Downtime	0-3.5	0-6	0-9	-
Employee Fluctuations	-	-	-	-

Tab. 6.10: Telco scenario QoS interval rules for TI data

The table is read in the same fashion as described in the previous section. In contrast to aerospace scenarios, the rule set favours operational TIs. These very frequently observed TIs capture an organisation's short term trustworthy behaviour as needed in a rapidly developing VO environment, while shifting down the influence of long term behaviour for the reputation computation.

The intervals and their interpretation with respect to the agent classes is the same as described for aerospace scenarios:

- $[1, 0.7[$ - high quality and exceptionally well performing agent
- $[0.7, 0.5[$ - well performing agent
- $[0.5, 0.2[$ - poorly performing agent
- $[0.2, 0]$ - exceptionally poorly performing agent

Also the same is the stochastic mapping of the agent classes 1 and 4 to the intervals. Since this section evaluates a telco scenario setting, the mapping of agent classes 2 and 3 is reversed, putting telco specialised agents of class 2 in the higher interval and the aerospace specialised agents into the lower.
As in the aerospace scenario, STORE's ability to separate the agent classes in a telco scenario simulation is first analysed. First on the basis of the generalised reputation value R, again serving as the worst case for an uninformed VOM. The average case occurs when telco specific trust preferences are applied and the resulting metric ψ_{AH} is computed for decision support.

These cases are first analysed by regarding for each the separation of the polarised agent Classes 1 and 4 and the the separation of the specialised agent Classes 2 and 3. A one-sided t-Test with the same setting as above (Niveau $\alpha = 5\%$) is applied and the following hypothesis, in negative form, are used:

H_{14}: STORE is not able to differentiate between agents of Classes 1 and 4.
H_{23}: STORE is not able to differentiate between agents of Classes 2 and 3.

More concretely, the equality of the mentioned figure's, R and ψ_{AH}, expectation values is tested. Table 6.11 lists the resulting p-values. The p-values denote the probability with which the respective hypothesis can be accepted.

Hypothesis	average case	worst case
H_{14}	$5,38879 \cdot 10^{-68}$	$3,42637 \cdot 10^{-90}$
H_{23}	$2,59943 \cdot 10^{-49}$	$0,00102792$

Tab. 6.11: t-Test results in the telco scenario

Also in the telco scenario, the very encouraging observation can be taken that the p-values, for all hypothesis in all cases, suggest to deny the hypothesis with high significance. This result is even higher appreciated than in the aerospace scenario, since the agile telco scenario environment makes it more difficult to distinguish especially between Class 2 and 3 agents. The QoS figures of agents in a telco scenario (Figure 6.8) compared with agents in an aerospace scenario (Figure 6.9) show the small peaks, changes in an agent's behaviour due to an emphasis of the short term development that is subject to more changes and therefore e.g. more frequent TI observations of the Operational TC. Agents of Class 2 and 3 exhibit even more similar performance, leading sometimes to intersections of their QoS measures which does not happen in the aerospace scenario. This fact helps to interpret the highest p-values across both scenarios in H_{23} worst case $(0,00102792)$. Distinguishing between the specialist agent classes is most difficult in this worst case, but with STORE's reputation based decision support, it is still possible with high significance. Another outstanding result is the steep improvement in the same hypothesis's average case to a value of $2,59943 \cdot 10^{-49}$. Again in this scenario, by applying a comparably simple metric as ψ_{AH}, an informed VOM receives even better decision support from the STORE reputation system.

The following Figures show the results of the agent welfare analysis. This analysis is measured by the agent's cash rates, in turn based on the reputation independent QoS values.

Figure 6.17 is dedicated to the random matching benchmark scenario, executed with telco scenario settings. Since even well performing agents are randomly matched with inferior ones, nearly constant cash rates around zero for the better performing ones are to be expected. More surprising is the constantly declining cash rate of the aerospace specialised Class 3 agent that nearly matches the Class 4 agent's decline. This fact can be ascribed to the construction of the telco QoS rule set. It only emphasises TIs from the Operational TC besides the generalised reputation. Since a Class 3 agent exhibits only a mediocre performance in such TI, only lesser QoS values are computed for these agents, resulting in low pay-offs even when matched into VOs.
Telco scenarios supported by STORE's reputation based decision support are depicted in the worst case in Figure 6.18 and in the average case in Figure 6.19. The worst case is again defined as a VO scenario with an uninformed VOM who expresses his trust preferences as the ones suitable for the diametrally opposed

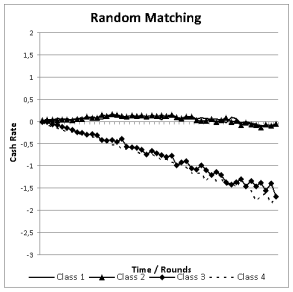

Fig. 6.17: Cash rates for the four agent classes in a random matching telco scenario.

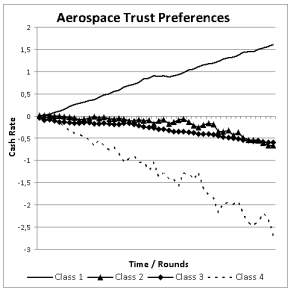

Fig. 6.18: Cash rates for the four agent classes in a telco scenario with wrong trust preferences.

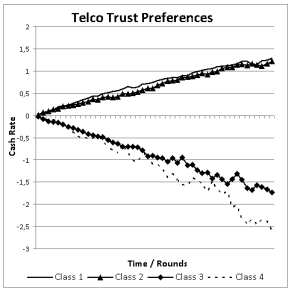

Fig. 6.19: Cash rates for the four agent classes in a telco scenario with correct trust preferences.

VO class. In this case, expressing aerospace scenario trust preferences ($\omega_{CE}^{\vec{}}$) in a telco scenario. In the average case, the correct trust preferences $\omega_{AH}^{\vec{}}$ are expressed by an informed VOM. In both diagrams, the boundaries of achievable cash rates are set by the uniformly well and badly performing agents of Classes 1 and 4. In the worst case, the aerospace trust preferences emphasise the Financial and Environmental TCs, but not the one relevant for a telco scenario, the Operational TC. Agents of Class 3 are therefore slightly preferred by STORE's reputation based decision support, they are matched more frequently into VOs. The real performance of Class 2 agents still excels in their operational aspects, resulting in higher QoS values and higher pay-offs, compared to Class 3 agents. However, since they are not matched into VOs all the time, Class 3 agents gain ground by being matched more often, resulting in the observable comparable, though slightly declining overall cash rates of Class 2 and 3 agents. Since the aerospace specialised agents do not behave maliciously, but aim at delivering service of their highest achievable quality in all TCs, a VO consisting of such agents does not automatically fail. Due to possible mishaps in operational aspects, slower delivery or less speedy response to customer complaints, may reduce the VO's profit margin.
The average case looks very well again. By configuration, the telco scenario can be formed, consisting only of the VOM and Class 2, telco specialised agents, as members. This explains the visible results in the right hand chart. With the telco trust preferences, STORE's reputation based decision support allows to resolve Class 2 and 3 agents reliably all the times. Class 2 agents are continually selected as the most trustworthy members for the formed telco scenarios, their cash rate rivaling the one of a Class 1 agent. In consequence, Class 3 agents show a cash rate similar to Class 4 agent.

The last two subsections evaluated STORE's reputation based decision support capabilities for the partner selection in concrete VO contexts. The obtained results are very supportive to the main research question stated in this thesis if STORE can well provide such support for all classes of VOs. Comparisons between the VO scenarios were mentioned in several paragraphs already, but no comprehensive comparison is drawn so far. The following paragraph intends to close this gap. A comprehensive comparison between the opposing VO scenarios is difficult due to their heterogeneous properties. The biggest difficulty is introduced by the varying lifetimes. The scenario settings reflect this in the different numbers of played rounds (20 aerospace and 50 telco), balanced by the varying real time representation of one round (years versus days). The smallest common denominator allowing a comparison is therefore one round, representing a full VO lifecycle. The well suited key figure for such a comparison then measures the impact of STORE's reputation based decision support on the VO environments without relying on the reputation ratings themselves, since

the application of trust preference vectors makes these ratings a difficult target for a direct comparison. The VO production rate, which can be measured per round, is a better suited, reputation independent, key figure for such a comparison. Figure 6.20 illustrates the visual results.

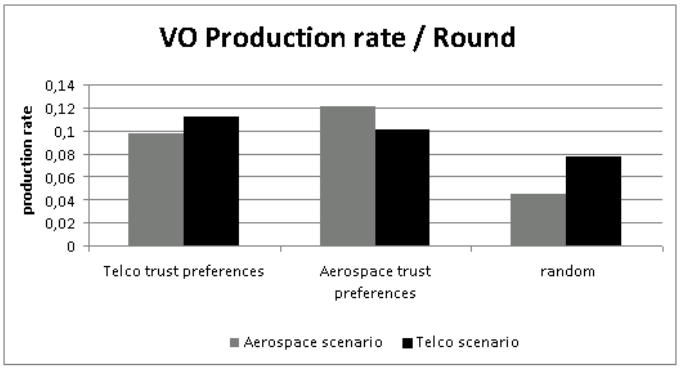

Fig. 6.20: Overall production rate per round over all agents for the two VO scenarios with different trust preferences

The light gray bars represent simulation results run with the aerospace setting, the black bars represent a telco setting. To also take the VOM expressed trust preferences and the benchmark scenario into account, the chart is divided into three groups of bars. On the left side, the scenarios with telco trust preferences are shown, in the middle with aerospace trust preferences and the random matching benchmark scenario to the right. The scenarios are compared by the average VO production rate, normalised to one round. As a first outstanding observation, STORE's reputation based decision support is preferable to none in all scenarios, even under worst case condition. The random matching scenario's production rates are in all cases inferior to the STORE supported ones.
In the STORE supported scenarios, the worst cases with wrongly expressed trust preferences still allow the simulated VO environment to achieve a large fraction of the average case's production rate. This supports the previous interpretations that VOs forming under such worst case conditions are not automatically destined to fail, but will suffer profit losses. This also emphasises the role of an informed VOM. The VO's business opportunity can be better exploited, if the corresponding VO context is already correctly expressed as the VOM's trust preferences during the VO's formation phase.

In all evaluated simulation scenarios so far, the agents behaved rather statically. Depending on their class assignment, they performed according to a class defined quality. Dynamic behaviour could only be observed due to the stochastic nature of the agent's actual behaviour, set by their TI data. The following subsection is now going to change this and analyses STORE's ability to reflect an agent's dynamic changes in trustworthy behaviour.

6.3.4 Sensitivity Analysis

Many reputation systems (and their MAS approaches for evaluation), even sophisticated ones also taking a stochastic approach such as TRAVOS[113] or Roger's approach using Dirichlet distribution modeling [93], have difficulties with dynamic behaviour of their trustees. In case of TRAVOS, this is clearly specified as future work. Their agents can not follow a dynamic strategy, only "noisy" behaviour based on random variations is allowed. It is one of STORE's strengths that it can, by its design, reflect the agent's dynamic

behaviour in its computed reputation. To allow for an evaluation, the SF allows to define agents that can change their goal oriented, not random, behaviour from one round to the other. The following scenarios define several conditions in a VO environment where an agent, radically or slightly, changes its behaviour. STORE's reputation based decision support should reflect this change in an appropriate, VO context specific time interval for a VOM to base his decisions on the updated reputation. In agile, fast paced VO environments, e.g. in a telco scenario, this time interval should be shorter, while long-lived aerospace scenario environments emphasise a member agent's long term trustworthy behaviour, asking for a larger time interval.

Scenario "Sudden Change"

In the first scenario, a sudden change in an agent's behaviour is defined. Table 6.12 lists the setting.

Parameter	**sudden change**
# of VOM agents/class	(1/0/0/0)
# of M agents/class	(4/0/0/0)
# of M per VOM	3
# of rounds	20
# real time/round	years
# of blind rounds	2
Initial endowment	0
Transaction cost	0.15
# of repetitions	1

Tab. 6.12: Simulation parameter in the "sudden change" scenario

The scenario includes one VOM, seeking three members for its VO. Four member agents are available. All are from the same agent class (here, Class 1), because different class assignments would introduce side-effects into the partner selection, based on class specific differences in the agent behaviour. These differences would make an analysis of the partner selection process with the sudden agent behaviour change impossible. Since the reputation supported partner selection process in this particular scenario is of interest, the raw results are analysed and no average figures from repeated executions of the scenario.
The scenario is defined on purpose that always one member is not selected for the VO. The agents, numbered from 0 to 4, begin the simulation with the same reputation values. Therefore, the members are at first selected with equal probability. After several rounds, the member agents 0 and 1 suffer a large performance decrease, while at the same time the remaining two agents suffer a smaller decrease. The period of decreased performance lasts for four rounds, then, the agents revert to their previous performance. Figure 6.21 shows the results, a chart with the agent's generalised reputation values over the rounds to the left and a table with the selection results to the right. The scenario is executed without a specific VO context, no trust preference vectors are supplied. The table denotes the four agents as the columns and the rounds as the rows. A darker cell colour signifies that the particular agent is matched into the VO for this round, a lighter colour signifies the agent was not selected in this round.

Up to round 10, the member agents are selected randomly for the VO due to their (nearly) equal generalised reputation values. After the performance decreases, agents 3 and 4 are always selected since they only suffered a smaller performance decrease. Theoretically, one of the agents 0 and 1 should randomly be left over from the selection process, since both agents suffered the same large decrease. The fact, that in this simulation run agent 0 is always left over has to be attributed to chance. The generalised reputation

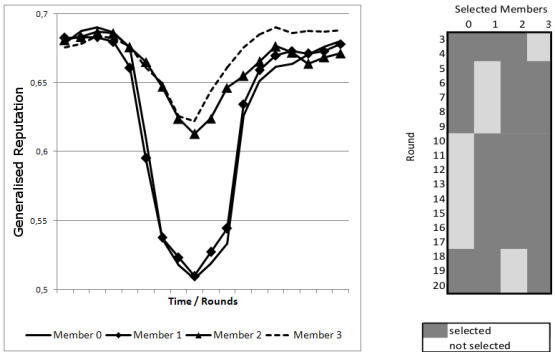

Fig. 6.21: The reputation values for four potential members changing their quality over time and a table indicating which agents are selected for VO membership.

values in the left chart underline the interpretation. The values quickly adapt to the changed behaviour with a smaller decrease for the agents 3 and 4 and a larger for the others. When the agents revert back to their initial performance, the generalised reputation values quickly follow. It is important to notice that an agent's behavioural changes do not induce any kind of lengthy oscillating into the reputation values which would require a tune-in phase. At most one overshoot is sometimes visible, e.g. in the case of Member 3 in round 14.

This first scenario, serving as an initial sensitivity analysis, confirms STORE's desired behaviour. STORE's computed reputation ratings quickly adapt to changes in an agent's trustworthy behaviour. The graphically represented reputation values suggest a comparably quick adaptation, an in-depth analysis of the reaction speed is provided in the following scenarios.

Scenario "CE Agent Change"

This scenario is set in an aerospace scenario context. The setting is modified from the above one in Table 6.7. Instead of the original member agent mix, the scenario consists of two new agent classes, derived from the Classes 1 and 4 to behave dynamically. The first Class 1' - "positive change" improves its performance after four rounds. The Class 4' - "negative change" decreases its performance at the same time with the same rate. In more detail, every agent begins with the identical configuration as its base agent class. In a later simulation round, a dynamic component, e.g. faster service delivery, is introduced into the agent's behaviour. It then acts differently compared to its base agent class. The dynamic component is described in detail for each sensitivity analysis simulation. In this "CE Agent Change" simulation, three agents of the first class are configured and four of the second. All agents begin again with the same generalised reputation values. During the first four rounds, the agents are therefore selected for the VO with equal probabilities. Figure 6.23 illustrates, using the QoS measure, the described change in the behaviour of both agents.

Figure 6.22 shows the corresponding generalised reputation value development. One can observe, that STORE reacts to the change already in the following round. It reaches the new reputation value, reflecting the current behaviour, already two rounds after the change. In this scenario, the same configuration as in the previous ones is applied to the STORE reputation system.
With this fast reaction to the agent's changes, STORE's reputation based decision support starts already

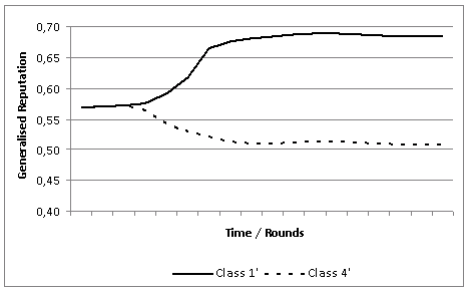

Fig. 6.22: Generalised reputation value in the scenario "CE agent change"

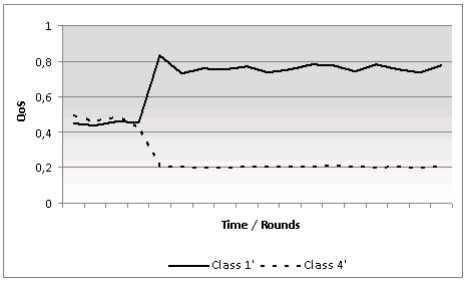

Fig. 6.23: QoS in the scenario "CE agent change"

giving the appropriate advice to the next VOM, who is interested to form a VO with the observed member agent. Technically, the computed reputation already starts to adapt within the round, upon the first TI data observation after the agent's behavioural change. The SF's results can only make this change in the agent's reputation visible in the following round, since the framework is designed to evaluate STORE's reputation based decision support during the VO's formation phase. STORE recognises these behavioural changes, but the SF does not query for an agent's reputation during later VO phases.
The impact of this decision support on a VO environment, measured again in the involved agent's cash rate, is depicted in Figure 6.24 for the agent who changed to the positive, in Figure 6.25 for the agent who changed to the worse.

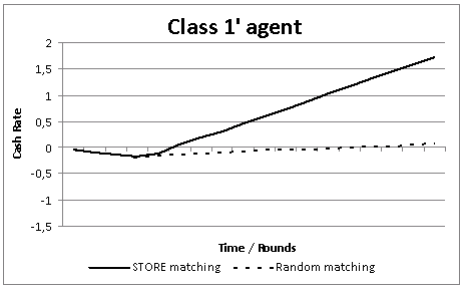

Fig. 6.24: Cash rate for a Class 1 agent, generated by STORE based- and random matching

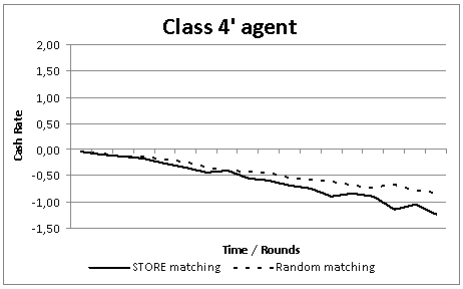

Fig. 6.25: Cash rate for a Class 4 agent, generated by STORE based- and random matching

Both Figures also include the results of the same benchmark scenario, where agents are randomly matched. The Class 1' agent's cash rate rises steeply, after his positive change in performance, while the Class 4' agent's cash rate declines even worse than in the benchmark scenario. After the changes in behaviour, the Class 1' agents are now the preferred VO members in the STORE supported scenario. This also implies, that the Class 4' agents are no longer selected, while they were at least randomly selected in the benchmark scenario, with low pay-offs.
This very positive result of STORE, properly handling dynamic changes in an agent's behaviour, is VO scenario independent. The scenario is therefore not repeated under VO specific settings.

Scenario "Reaction Speed"

The previous scenario in the sequence of simulation scenarios, dedicated to STORE's sensitivity analysis, started to look into a quantitative analysis of STORE's reaction speed to changes in an agent's behaviour. This scenario follows up on that topic, evaluating STORE under two configurations within the same simulation setting. The configurations are chosen because they affect the reputation system's reaction speed, resulting in faster and slower reaction.
Evaluating STORE's reaction speed is especially important when re-calling, that its decision support should serve any class of VO. VOs can have hugely varying life expectancies that pose different demands on how quickly a reputation rating should reflect a trustee's change in behaviour. Ad-Hoc Service Provisioning VOs live for at most days. There, the most recent development in a potential VO member's trustworthy behaviour is of higher importance than more long-term trust aspects. A VOM may quickly enter into another VO with the same partners after the last one dissolved, or even multiple VOs with the same partners in parallel. A Collaborative Engineering VO on the other hand typically lasts for years. Throughout such a VOs lifetime, its partners inevitably exhibit changes in their behaviour. But what counts more in the end are their mid to long-term oriented trust aspects, how well they can recover e.g. from set-backs and still contribute reliably to the VO, regarded over its entire lifetime. In summary, telco scenarios demand from the STORE reputation system to react quicker to behavioural changes than aerospace scenarios do. This requirement implies, that a trustee's recent behaviour is more important for a quicker reaction than past behaviour.
STORE bases its reputation rating on observable properties, characterising an organisation's trustworthy behaviour. These properties are called the TIs, which are observed periodically in TI specific regular time intervals. For the MAS based evaluation, the SF takes care of these TI observations that define an agent's behaviour. Each TI carries an attribute that defines the time interval Δt_{obs} (see Subsection 3.1.2), how far back data of this TI instance is considered for the current reputation computation. To alter STORE's reaction speed, setting Δt_{obs} is the most important configuration parameter. A smaller interval emphasises on more recent behaviour, STORE reacts more quickly, while a larger interval results in slower reaction time.

Since a simulated VO is only an abstraction of a real world VO, a model designed to evaluate STORE's reputation based decision support, it is meaningless to artificially set absolute lifetimes. Since the VO simulation scenarios are supposed to deliver key figures that are compared to evaluate STORE's performance, it makes more sense to list relative times, relative to a VO lifecycle, one simulation round, in an application domain context. In this "Reaction Speed" scenario, a similar setting to the first scenario "Sudden Change" is used (Table 6.12), but only one type of behavioural change is analysed. STORE computes the reputation ratings with two configurations, observing the implemented TIs in different periods, as defined in Table 6.13. The cell values denote, how many past observations are incorporated in a TI update.

TI	fast reaction	slow reaction
Country Bond Spread	2	20
Cash Flow Margin	1	10
Complaint Rate	5	50
Delivery Delay	5	50
System Downtime	10	100
Employee Fluctuations	1	10

Tab. 6.13: Δt_{obs} for the TIs in "slow" and "fast reaction" setting

The relative differences between the TIs amount due to their real-world differences, in which time in-

tervals new data can be observed. While EF and CBS are only rarely observed, SD is nearly constantly monitored. The difference between the configurations is chosen that for a slow reaction, 10 times more past data observations of each TI are considered.

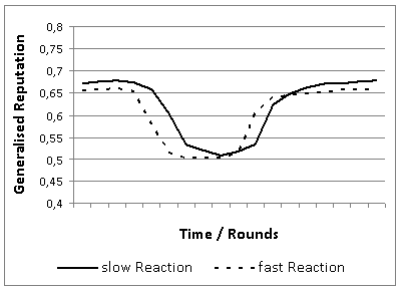

Fig. 6.26: Reputation reaction on quality change with STORE configured for slow and fast reaction.

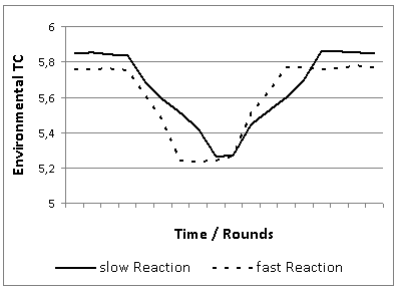

Fig. 6.27: TC reaction on quality change with STORE configured for slow and fast reaction.

Figure 6.26 shows the results for the generalised reputation value R. In round four, the regarded agent drops suddenly in performance. While the quickly reacting STORE reputation system adapts the reputation value already in the next round, the slower configuration requires two rounds. It is not to be expected, that an increase in the agent's behaviour would lead to different results, but the increase is simulated, with the same reaction time results, as well. It is also interesting to know, if TIs from all TCs contributing to the generalised reputation value, behave in the same fashion or if one exerts a dominating effect. Figure 6.27 depicts, as an example, the results obtained for the "Environmental" TC. This TC, encompassing only rarely observed TIs, follows a similar chronological behaviour as the generalised reputation value. Other TCs show similar results, they are omitted here since nothing new can be learned from them.
Interpreting this in a VO context leads to further beneficial results for STORE's reputation based decision support. Agile telco scenarios benefit from a "fast reaction" configuration setting, where misperforming agents are already recognised in the next round, when the following VO forms. The VOM then can immediately select the most trustworthy VO members, based on their most recent behaviour. In an aerospace scenario, a "slower reaction" setting is more appropriate, emphasising the longer term trustworthy behaviour of the members. Of course, no configuration setting should tend to extreme reaction settings, since this would lead to over-emphasising or entirely disregarding the most recent behaviour. A balance should always be kept. Besides the TI's observation time interval, other TI attributes can be altered as well to achieve a certain desired behaviour. The weighing function for instance is another parameter that can be altered to emphasise behaviour from a certain time without further reducing or increasing the observation time interval. These TI attributes are described in Subsection 3.1.2.

6.3.5 Attacker Simulation

The attack classification analysis in Chapter 5 underpins STORE's resilience against many attacks, other reputation systems suffer from. This resilience is rooted in STORE's novel trust model and architecture. However, the analysis also revealed a possible susceptibility to a certain kind of freeriding attack. The attack, classified as A3 in Section 5.2, is simulated with an attack specific setting in this Subsection to evaluate, how STORE copes in such a situation.

A typical freerider tries to hide information, e.g. about his lower trustworthiness, to receive increased profits at a later point in time. When a reputation system keeps track of a freerider's performance, he may for instance perform reliably as expected in less crucial situations, e.g. in low volume transactions. But when an important long running, high volume transaction is about to be conducted, his performance drops to increase his profit. A clearly malicious freerider, drastically dropping in performance, e.g. exhibiting a sudden increase in his delivery delays or not answering to complaints any more, is very likely to be detected, since STORE will quickly reflect this change in his recent reputation ratings. Such a behaviour would appear similar to the simulations conducted in Subsection 6.3.4. A more successful approach to a freeriding attack might be only a slight, but sustained drop in performance. Such a freerider will aim at "staying under the radar" of decision makers, he will be still be selected for VOs since his reputation values are not significantly smaller than those of honest competitors.

Such an attack may for instance happen under the prerequisite of a STORE reputation system instance, that is used for more than one VO application domain. Such a case may occur due to cost saving measures on the system's owner side, where only one system is maintained with an appropriately generalised system configuration, only relying on the VOM's trust preferences for the VO context. It is assumed that the freerider offers services, e.g. electronic content or storage services, that are in demand in both domains. But he is only specialised in one domain and therefore less reputable in the other. In such a scenario, a freerider may appear as an honest participant due to his earned reputation rating in his VO application domain, but tries to enter VOs of other domains as well, where he is maximising his profits on account of his reduced performance.

Parameter	**CE Scenario**	**Telco Scenario**
# of VOM agents/class	(1/0/0/0)	(1/0/0/0)
# of M agents/class	(2/1/3/0)	(1/1/1/0)
# of M per VOM	5	2
# of rounds	20	50
# real time/round	years	days
# of blind rounds	2	2
Initial endowment	0	0
Transaction cost	0.15	0.15
# of repetitions	50	50

Tab. 6.14: Simulation settings for both freerider scenarios

The simulation setting for the freerider scenario in Table 6.14 reflects these prerequisites. With exception of the amount and ratios of agents, the aerospace and telco scenario settings remain the same. The freeriding agent belongs to the agent class of VO specialised agents of the opposite application domain than the currently simulated one. For instance, if a freerider is simulated in an aerospace scenario, the freerider belongs to Class 2 of telco scenario specialised agents. This agent class assignment ensures that the freeriding agents appears as an honest agent to VOM agents looking for potential VO members.
Instead of simulating a healthy VO environment, as in above scenarios, to analyse its development with and without STORE's reputation based decision support, the amount of agents is carefully chosen to evaluate STORE's reputation decision support for a single VOs. Freeriders are inserted in both, VOs of the aerospace and telco application domain. Since the repeated lifecycle of single VOs is simulated, only one VOM is configured for each setting. Then, the amount of member agents needed to form a VO in the analysed application domain is set, plus one, the freerider. The honest member agents belong to equal fractions to Class 1

and the class of agents specified in the current application domain, Class 2 or 3. With this setting, the VOMs are able to form a VO in each simulation round, but they are not forced to select the freeriding agent. Since the computed matching is entirely based on STORE's reputation, the reputation based decision support has a more positive impact on the partner selection during the VO's formation, the fewer times the freerider is matched into the VO. It is expected that Class 1 agents are always selected, due to their overall positive performance. They are put into the mix to benchmark the best possible outcome in each round. The first simulation scenarios, e.g. Figure 6.7, show that a freerider will have more chances of success, the nearer he gets with his reputation rating to honest agents.
Since the freeriding scenario is evaluated in a VO application context, the metric ψ based on the VO specific trust preference vector ω, with the same values as above, is applied.

First, a freerider is evaluated in an aerospace scenario. Freeriding in this setting is more likely, since it is more profitable for an attacker, compared to other application domains. Mounting a successful freeriding attack in an aerospace scenario implies that the attacker profits from high volume business transactions and not micro payments as e.g. with electronic services in the telco scenario's domain. Furthermore, a successful freerider in an aerospace scenario can benefit from a sustained attack, due to the VO's high expected lifetime. Such profits may even be enough of an incentive for newcomers or members of other agile, short-lived VO application domains to enter foreign domains to mount an attack. They hereby consciously take the risk of being put permanently out of business, if ever caught.

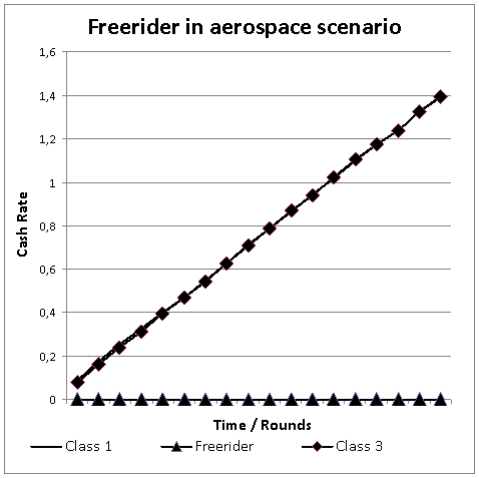

Fig. 6.28: Agent cash rates in an aerospace scenario with freerider.

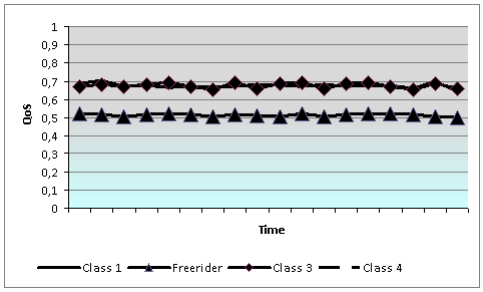

Fig. 6.29: Agent QoS values in an aerospace scenario with freerider.

Figure 6.28 plots the simulation results as the cash rate figures of representatives of the agent classes. For comparison, the QoS measure is plotted in Figure 6.29. The Class 1 and 3, aerospace specialised, agents perform at nearly the same high QoS level, while the freerider's level, though staying within the same QoS interval, is slightly less. The left hand chart shows basically equal cash rates for the agents of Class 1 and 3, while the freerider's rate is zero. This means, that the freerider is never selected and matched into a VO. Only the most trustworthy agents transact, receiving pay-off, while the freerider never transacts and therefore not even pays the transaction fees. This very good result is also achieved due to STORE's ability to take

a VOM's trust preferences into account. Here, an informed VOM expresses his preferences for financial and environmental trust aspects, the vector ω_{CE} emphasises these TCs. The freerider, only after the short-term benefit, does not appear reputable due to his TI observations in the Financial and Environmental TIs, which are quickly captured by STORE's reputation. Therefore, STORE's reputation based decision support works in the best possible way, the VOM never selects the freerider for one of his VOs.

It is considerably less profitable to mount a freeriding attack in a short-lived and dynamic telco environment, but not impossible. On the one hand, since freeriders actually deliver a service, but of lesser quality or with interruptions, such an attacker must participate more frequently in transactions to gain from low volume transactions. On the other hand, freeriders can hide better in such agile environments. Their fluctuations in trustworthiness will be best detectable in frequently observed TIs, such as from the Operational TC. But also regular fluctuations, e.g. due to a "blue monday" effect, are expected to show up in such observations, which makes the task to single out a freerider simply by reputation more difficult.

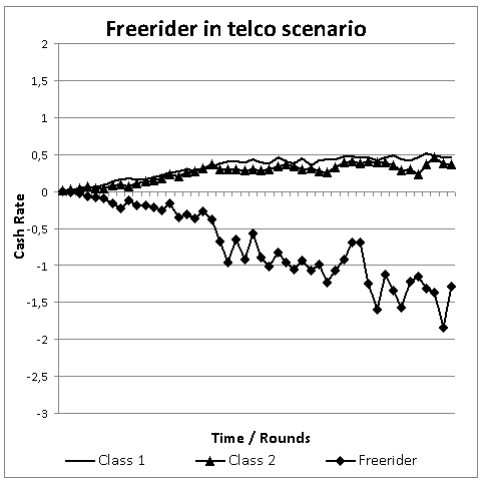

Fig. 6.30: Agent cash rates in a telco scenario with freerider.

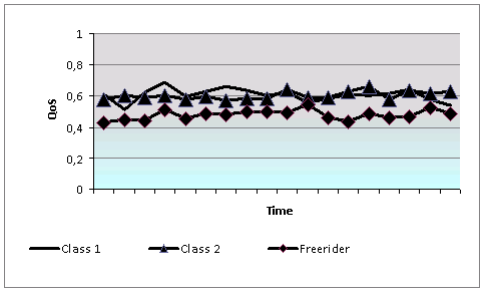

Fig. 6.31: Agent QoS values in a telco scenario with freerider.

Figure 6.30 depicts the cash rates of the different agent representatives and Figure 6.31 the corresponding QoS values. The starting basis is similar to the aerospace freeriding scenario, the telco specialist Class 2 agent performs nearly as well as the Class 1 agent in terms of QoS measurements. The freerider performs below, but also slightly less differentiated as in the aerospace scenario. When looking at the cash rates, it first becomes obvious that the freerider is from time to time selected for the telco scenario. This occurs on account of the Class 1 and 2 agents, whose cash rates are only slowly increasing. But the Class 2 agent is considerably better off than the freerider. Due to the dynamic telco environment with QoS measures emphasising the agent's short behaviour based on frequently observed TIs, such a result was expected. The question now is, which impact the occasional selection of the attacker, even implying an informed VOM expressing the appropriate trust preferences, has for the VO environment as a whole. The honest Class 2 agent suffers from the near zero cash rate, but at least does not decrease into negative numbers. This does the freerider, continually. He therefore has no incentive to continue with his attack, since no profit comes out of freeriding. Without any other motivation, the freerider will quickly abandon the attack. The grade, honest agents suffer from such attacks, can also be influenced by other means. A VO environment demanding less

”transaction fees” for instance would raise such a VO member’s income.

The simulations showed that STORE also performs well when under attack. Especially in an aerospace scenario, where such freeriding attacks are more likely to happen and can do more damage, the attacker was continually singled out and never matched into a VO. Dynamic, short lived telco scenario environments pose more of a challenge, the freerider is occasionally selected, but due to a continually decreasing cash rate, has no incentive to continue this financially motivated attack over a longer period of time.

6.3.6 Comparison with the Beta Reputation System

In this last evaluation Subsection, the STORE reputation system is compared with the Beta reputation system. Reputation systems based on a lean, homogeneous trust model, e.g. feedback based systems with an integer scale for feedback values, are frequently compared with each other in simulations. Comparing STORE in contrast is a much more difficult task, due to its rich and heterogeneous underlying trust model. STORE computes reputation from the heterogeneous, stochastic TI model. Also other factors make such a comparison difficult. Many reputation systems and their reputation mechanisms are designed for individuals or other kinds of constrained objects as trustees, while STORE is designed to cover complex organisations as trustees. Nevertheless, the exercise of comparing STORE with the Beta reputation system is undertaken and the results are carefully interpreted.

The Beta reputation system [62] is described in Subsection 2.1.2 in more detail. Agent reputations are computed from received feedback about conducted transactions. The reputation mechanism has several implications when using the Beta reputation system for reputation based decision support during the formation of a VO:

- VO partners must be willing to provide honest feedback.
- The computation of a useful reputation value requires a considerable amount of transactions taking place.

This makes the Beta reputation system more suitable for agile telco scenario environments.
As reputation mechanism, Jøsang et al. use the Beta distribution, see Appendix A.1.4 for details.

$$Rep(r,s) = E(\varphi(p|r,s)) = \frac{r \cdot s}{r+s+2}$$

The reputation value Rep is computed as the expectation value of a Beta distribution φ with parameters r and s. Feedback is given in binary form, r maintains the amount of positive transaction outcomes with a partner, s the negative ones. STORE uses the interval [0, 1] for many key figures. To compare it with the Beta reputation system, a compatible and comparable source for feedback must be defined. This implies, that it fits the formal input requirements for binary feedback and is STORE reputation independent to make a comparison possible in the first place. An agent’s QoS value is the best candidate for such a source. The QoS value is explicitly introduced as a reputation independent, agent specific performance measure. For simplicity’s sake, a QoS value in $[0, 0.5]$ denotes a negative transaction outcome, $]0.5, 1]$ a positive one. In each simulation round, feedback derived from each agent’s QoS value is submitted to the Beta reputation system, simulating honest feedback. The Beta reputation value Rep and STORE’s generalised reputation value R can then be compared with each other. Feedback, as used by the Beta reputation system, originates from participants of past transactions that occurred in a certain application context. In contrast to STORE’s ability to make such a context explicit by means of a preference vector and an application scenario specific

configuration, feedback only implicitly carries a context. The generalised reputation value presents itself as a neutral, unbiased measure to compare with the Beta reputation system. Since R behaves identical in telco and aerospace scenarios [14], one simulation suffices for this comparison.
The following Figures show the results for the four agent classes, along with each agent's QoS value developed over the simulation rounds. Rep and R are drawn with different scales in these Figures.

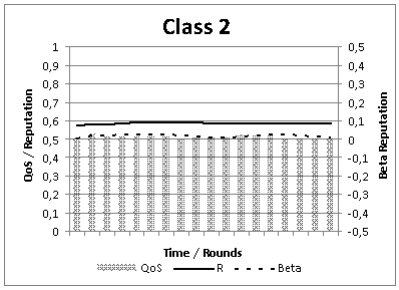

Fig. 6.32: Class 2 agent QoS, reputation values ("R") from STORE and from the Beta Reputation System ("Beta") in an aerospace scenario.

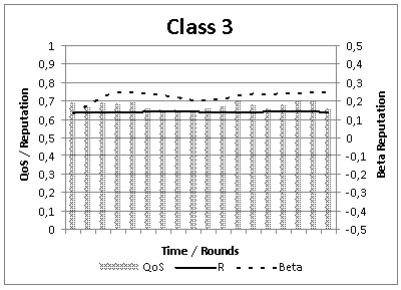

Fig. 6.33: Class 3 agent QoS, reputation values ("R") from STORE and from the Beta Reputation System ("Beta") in an aerospace scenario.

On a first glance, both reputation systems meet the expectation to deliver reputation based decision support during the formation of a VO. Technically, the Beta reputation system suffers more from tune in effects than STORE, e.g. constantly visible in Figure 6.34, where a Class 3 agent shows more stochastic variation in its TI data than usual. An increased sensitivity for the Beta reputation system is to be expected.
In Figure 6.34, Rep also shows a tune in effect when struggling to reflect a Class 4 agent's reputation.

The main differences between the Beta reputation system and STORE lie in the designs. While feedback mechanisms are easily exploited, e.g. by malicious or frequency manipulated feedback (see Chapter 5), STORE is designed to rely on TI data observations, taken in regular intervals and difficult to manipulate without detection. Such data observations are taken in regular intervals, even when the trustee is not involved in any business transactions. STORE's reputation rating is therefore always up-to-date, while the

[14] assuming the same STORE system configuration which would be required for comparing across scenarios.

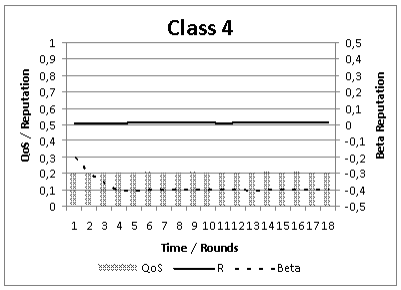

Fig. 6.34: Class 4 agent QoS, reputation values ("R") from STORE and from the Beta Reputation System ("Beta") in an aerospace scenario.

Beta reputation needs to tune in again after a period without having the trustee being involved in transactions. Feedback is also provided ex post, after a transaction took place. STORE is not limited in its TI observations, these are taken ex ante, during a possibly long running transaction, and also ex post. Missing or defective data is also taken care of by the Bayes update. The Beta reputation system delivers with the single Rep value in $[-0.5, 0.5]$ the probability of the next transaction outcome. This approach does not capture VO application domain specific trust aspects, such as e.g. a higher probability of trustworthy operational behaviour for the next one, but not financially. With STORE's trust preference vector, capturing such an application context is possible.
The advantage of the Beta reputation system on the other hand is its ease of integration into different application scenarios. The maintenance costs of such a system are not as high as STORE's, with respect to computational resources.

Table 6.15 lists the results of the comparison.

	STORE	Beta Reputation System
computational resource requirements	high (Bayes Net)	low (beta distribution)
input data	very flexible, TI observations (extensible TI taxonomy)	binary feedback
reputation rating	reputation vector	value in [-0.5, 0.5]
VO application domain context	trust preference vector	-

Tab. 6.15: Comparison of STORE with the Beta Reputation System

6.4 Summary

This chapter successfully performed a MAS based evaluation of the STORE reputation system. The main research question, if STORE can provide automated reputation based decision support to a VOM for the partner selection during a VO's formation phase, can now be clearly and positively answered. Numerous

simulation scenarios delivered the proof, that STORE supports the selection of a subset of the most trustworthy partners for a VO from a larger set of potential partners:

(i) In the functional baseline, STORE was able to clearly and consistently distinguish between the four agent classes. A VOM is able to select the most trustworthy agents based on the reputation vector provided by STORE.

(ii) The next two simulation scenarios introduced the VO application contexts of an aerospace and telco VO. Applying application specific preference vectors, STORE proved to be able to distinguish among the agent classes by their reputation measure. This distinction, in both average and worst cases, was highly significant. By measuring the the simulated VO's production rate and by comparing with the random matching benchmark, a healthy, prospering market with the most trustworthy agents receiving the highest pay-off could be observed.

(iii) In a sensitivity analysis with agents changing their trustworthy behaviour, STORE could prove its strength in quickly reflecting those changes in the reputation measure. STORE's reaction speed can be configured, which is an important requirement for its application in different application scenarios, emphasising more on short or long-term trustworthy behaviour.

(iv) Revisiting the freeriding attack from Chapter 5, STORE proved to be resilient against this threat. The evaluation was conducted in both, aerospace and telco application context.

(v) Finally, the comparison with the Beta reputation system proved once more the suitability of STORE's design for VO environments. Technically, with respect to the reputation computation, STORE can slightly outperform the Beta reputation system in a fair, "worst case" comparison. Here, STORE did not play its strength of being able to integrate an application context through preference vectors and an application domain specific configuration.

7. CONCLUSIONS AND FUTURE WORK

In conclusion, this thesis presented STORE, a reputation system following a stochastic approach in trust model and architecture. STORE is able to provide automated reputation based decision support for a wide range of VOs. VO managers benefit from this support since they may rely on reputation as an additional measure for their decision making processes to select the most trustworthy business partners during the formation phase of their VO. The initiative for STORE's development originated from the observable trend in VO and related business environments where the increased pace requires a swifter set up of those environments. Another aspect of increasingly dynamic business environments are potential business partners business partners who enter and leave the environment with a higher frequency. VO Managers can no longer rely on personal experience with business partners alone since they share no common past with newcomers. STORE's reputation based decision support compensates for this lack of personal experience. It is well suited to be deployed into ICT environments following the SOA paradigm, which are expected to further fuel the demand for increased speed.
First, the following Section 7.1 summarises the previous chapters and reviews the research question, the contributions and results of this thesis. Section 7.3 then concludes with a brief outline of future work and related research questions.

7.1 Summary of Contributions and Review of Work

The main research question of this thesis addressed STORE's ability to provide automated reputation based decision support to a VOM for member selection during the VO's formation phase. This support is intended for the entire range of VO classes. By evaluating STORE in the two opposed aerospace and telco VO scenarios, which span the range VO classes, the question can be clearly positively answered.
Derived from this overarching research question, the thesis addressed several other more focused topics in detail:

- Can STORE separate specialised organisations of similar trustworthiness by reputation to provide decision support for a particular application domain? STORE is able to clearly and continuously separate such organisation by their reputation measure, VO Managers receive the decision support they require to select business partners specialised in their application domain.
- Can STORE cater for the dynamic trust aspects such as organisations changing their trustworthy behaviour over time? STORE is able to quickly reflect such behavioural changes in the organisation's reputation measure. Moreover, the reaction speed can be adapted to the needs of an application domain.
- Does STORE provide a robust reputation mechanism, which is resilient against attacks plaguing other systems, for instance eBay's? STORE's reputation mechanisms proves to be resilient against many

known attacks, as well as new ones trying to freeride on STORE's application domain specific decision support.

- How does STORE perform compared to a related reputation system, such as the Beta reputation system? Even neglecting STORE's ability to capture an application context, STORE performs better than the Beta reputation system. This fact considerable improves, when STORE plays its strength of explicitly serving for a specific application domain and configuring the versatile architecture for the domain at hand.

The objective of this work was to present STORE, the STochastic REputation system, its underlying trust model and the system's architecture. STORE is designed to provide automated, reputation based decision support to VO managers during their VO's formation phase. In this early lifecycle phase of a VO, the manager has to select a subset of the most trustworthy VO members from a larger set of potential VO members. The STORE reputation system is not restricted in its application to VO structures, but can be applied to other organisational structures having organisations collaborate.

The work presented in this thesis is comprised of the following contributions:

- It provides a novel trust model, taking a stochastic approach, for organisations that collaborate in VOs. This trust model is rooted in observable organisational properties that characterise trustworthy behaviour, the so-called Trust Indicators. Furthermore, the trust model takes recognised trust properties from literature into account, among others its subjectivity and trust being a directed, bilateral relationship.

- It presents design and architecture of the STORE reputation system that builds upon the novel trust model. STORE's architecture is well suited to be integrated into SOA environments as a reputation service. Its reputation based decision support mechanism is designed to support VOs set in arbitrary application domains and not just one in particular.

- It analyses STORE's resilience with respect to known attack vectors on reputation systems, leading to an attack classification for STORE in particular. These classes are analysed, revealing that STORE is by its design already resilient against the largest fraction of attacks, others are thwarted or become uneconomical with sensible system configuration. Configuration and system bootstrap suggestions, as well as other mitigation strategies are also provided.

- It provides design and architecture of a MAS framework for STORE's evaluation. The framework is capable to evaluate and compare STORE across VO application scenarios. Several simulation scenarios are defined, ranging from VO application domain specific, sensitivity analysis and attacker scenarios. In particular, STORE is successfully evaluated in the opposing VO application scenarios from Collaborative Engineering and Ad-Hoc Service Provisioning. Finally, STORE is compared with the Beta reputation system from literature.

The presented work was divided into several steps and proceeded along the following thesis structure. Chapter 1 provided an introduction into the thesis and introduced the document structure. First elemental definitions, e.g. of trust and reputation, were already provided in this first chapter. Chapter 2 was comprised of this thesis's theoretical foundations ranging from a related work discussion to a detailed description of two VO application scenarios. The following Chapter 3 presented all of STORE's aspects, that have to be dealt with before such a reputation system can be instantiated. These aspects are called design time aspects

and entail the trust model based on TIs, their detailed model and the taxonomy thereof. The Bayes Network design for the TI aggregation was also introduced. Chapter 4 continued with STORE's runtime aspects, addressing a usable system instance. This includes STORE's architecture and bootstrapping process. The chapter concludes with an analysis of the system's security, scalability and availability.
Then, Chapter 5 introduces an actor based threat model to perform a resilience analysis on STORE. The analysis yields an attack classification of possible attacks on STORE. In each case, a mitigating measure is suggested.
Chapter 6 dealt with STORE's evaluation, adopting a MAS methodology. A novel framework was introduced in the chapter's first half. The chapter's second half defined concrete evaluation scenarios and their setting to evaluate STORE in. The scenarios's simulation results were analysed and discussed.

In summary, these very encouraging evaluation results prove that STORE in fact is able to provide an automated, reputation based decision support to a VOM for VO member selection in all classes of VOs during their formation phase. Especially in challenging cases when business relationships in agile environments are entered with strangers, automated trust management as in the case of the STORE reputation system, can provide viable decision support to minimise the risk of selecting a "lemon".

7.2 Limitations

The main research question in this thesis asked if STORE is able to provide automated reputation based decision support to a VO manager during the partner selection process in a VO's formation phase. According to the VO classification in [32], VO classes do not differ sharply in their properties, they rather blend into each other seamlessly. For that reason, the evaluation of the STORE reputation system was conducted in the two opposed classes of collaborative engineering and ad-hoc service provisioning VOs.

The impact of STORE's decision support was specifically evaluated in the VO's formation phase, addressing business organisations. Selecting trustworthy partners in the beginning of the VO's lifecycle is a crucial moment to avoid costly partner replacement later on. For that reason, the evaluation's focus was put on the VO's formation phase. This does, by no means, prohibit STORE's application in other VO phases, related environments and observation of entities other than organisations.

During this thesis, the availability of data observations for Trust Indicators was not extensively discussed. This matter extends to a substantial topic of its own, easily capable to go beyond the scope of this thesis. TI data availability is likely to be met in hosted environments e.g. offering Software-as-a-Service (SaaS) solutions. With all necessary roles participating in the confines of an extended platform, a STORE reputation system instance can serve possible trustors and observe the trustee's TIs.

It is desirable to compare novel decision support systems with existing ones from the state of the art, which follow a similar approach. With respect to reputation systems, most state of the art approaches are based on the aggregation of, usually single dimensional, homogeneous trust measures, in most cases feedback values, towards a reputation measure. STORE is designed to explore the opposite approach, aggregating multi dimensional, heterogeneous trust measures, the TIs, towards a reputation value. The differences in the approaches makes a comparison of STORE with other state of the art reputation systems difficult. Furthermore, most systems lack a notion of an application context. Nevertheless, a comparison with the Beta reputation system is presented in Subsection 6.3.6, where those difficulties become apparent. Still, meaningful conclusions can be drawn from a system architecture perspective.

7.3 Future Work

When concluding the line of work that was conducted in this thesis, this does not imply it is finally finished and no longer used. Nowadays, in the reputation management research community, many new systems based on novel reputation mechanisms are invented, but work on only few of these sustainably progresses. Many approaches are abandoned after the project or thesis, they were developed in, finishes. Work on the STORE reputation system and trust model is supposed to continue, e.g. in publicly funded research projects. As guidelines, the following strands of possible future work are suggested.

7.3.1 STORE's design and architecture

A severe drawback of the application of Bayes Networks is the large computational cost for maintaining them. Currently, each TI update being injected into STORE's information nodes leads to a full recomputation of all CPTs in all nodes. Depending on the network's topology, it is however, under certain conditions possible, to exploit locality properties. This way, not every TI update would lead to a full network update, but only the CPTs of nodes in the vicinity of the updated information node would be recomputed, leading to more scalable STORE system instances.

STORE's BN currently only employs parameter learning. When new TI observations become periodically available, the TI is updated and the resulting posterior distribution is injected into the BN's corresponding TI node which triggers recomputations of each node's random variables. The BN structure though remains the same. The idea of structural learning could be employed to improve the calculation of an organisation's reputation measure. By applying structural learning, it could be possible to detect TIs that either decreased in relevance for the current VO, e.g. by incorporating direct feedback, or which just did not exhibit any changes in behaviour over a longer time period. The information nodes of these TIs can then be exchanged, according to a defined strategy or randomly, with other TIs, possibly of the same TC. The resulting updated BN structure is then compared to the previous one during a test period, where the one delivering better reputation measures is kept.

STORE's reputation interface, defining a metric Ψ on the inputs of trust preference and reputation vector captures the VO application context and makes it possible, to select the most trustworthy of a set of specialised, honest agents. This metric is currently the Euclidean metric, but effectively capturing an application context offers a huge potential of improved reputation based decision support. Improving STORE's reputation interface to capture a VO application context more effectively offers another strand for continuing this work.

7.3.2 STORE's application

In this thesis, STORE's evaluation focussed on the VO application domains of Collaborative Engineering and Ad-Hoc Service Provisioning. The results, especially from the sensitivity analysis showed that several system configuration settings can be adapted in a certain way, e.g. when defining the TI attributes or the preference node's trust levels, that the resulting STORE system instance provides an improved reputation based decision support for this VO application domain in particular. This gives rise to the expectation, that STORE profiles can be derived for VO application domains. Defining these profiles will require further work on applying STORE in other VO scenarios, e.g. eLearning VOs or Credit Union VOs.

In such a complex environment as in a VO, where automation is a clearly identified goal, reputation management should drive security and contract infrastructures and mechanisms.
Currently missing is a strategy and also technology to coordinate those mechanisms, e.g. a reputation system is not only providing trust values in the beginning, to set up a VO, but is also monitoring trust levels of VO partners during the entire VO lifecycle. In a set of background work that served as the inspiration for this thesis, Business Processes (BPs) on the top enterprise layer are foreseen as the integrative component allowing for the optimal combination and configuration of lower level service based reputation management with security mechanisms. Allowing for the BP to be controlled by reputation and security management in an automated fashion from the service layer, the BP model has to be extended. One possibility would be the injection of control elements into BPs. Such control elements alter the BP instance's control flow based on reputation measures from the BP instance owner's current collaboration partners. Since BPs are enacted during the VO's operational phase, the reputation support would not be limited to provide automated decision support during the formation phase but would be extended to the entire VO lifecycle. In [51], first work was published that designed and implemented such control elements as WS-BPEL [1] design patterns which could be injected into WS-BPEL executable BPs.
Executable BPs were chosen for this line of work due to several advantages:

- BPs are based on flexible designtime models and control elements can extend an existing process model with minimal intrusion.
- Focusing on executable BPs, such as processes modelled in WS-BPEL, the implicit use of a web service based SOA layer is already provided and enables a comparably simple integration of other subsystems, such as reputation management.
- BP models allow for dynamic runtime control of the process flow, e.g. branching, joining, based on reputation decisions.

These anchor points for future work hint at possible continuations of the work spent to create the STORE reputation system's trust model and architecture. They show that many interesting research questions can still be discovered in this area of trust and reputation management, justifying continued research work.

[1] Business Process Execution Language/WS-BPEL, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

8. GLOSSARY

AH	Ad-Hoc Service Provisioning, VO Scenario
ASP	Application Service Provider
BN	Bayes Network
BP	Business Process
CBS	Country Bond Default Spread, TI
CE	Collaborative Engineering, VO Scenario
CF	Cash Flow (Margin), TI
CR	Complaint Rate, TI
DAG	Directed, Acyclic Graph
DD	Delivery Delay, TI
DoS	Denial of Service
EF	Employee Fluctuation Rate, TI
EPS	Earnings Per Share
EVA	Economic Value Added
ICT	Information and Communications Technology
IT	Information Technology
MAS	Multi Agent Simulation
MRS	Marginal Rate of Substitution
PDA	Personal Digital Assistant
QoS	Quality of Service
RS	Reputation System
SC	Scenario Control
SD	System Downtime, TI
SF	Simulation Framework
SLA	Service Level Agreement
SMP	Stable Marriage Problem
SOA	Service Oriented Architecture
SP	Service Provider
STORE	STOchastic REputation system
TC	Trust Class
TI	Trust Indicator
TLS	Transport Layer Security
TP	Third Party
VO	Virtual Organisation
VOM	Virtual Organisation Manager
WLAN	Wireless LAN
WS	Web Service
WSS	Web Service Security
WSDL	Web Service Definition Language
XML	Extensible Markup Language

APPENDIX

A. APPENDIX

A.1 Mathematical Background - Stochastic

This appendix dedicated to stochastic background enumerates and describes the mathematical fundamentals used in the core of the STORE reputation system. Providing this section aims at a fully self-sustaining thesis giving the reader all the required background information to understand the contribution's concepts as well as its implementation and evaluation. The mathematical background encompasses distributions, density functions and related stochastic assertions for TI modelling.

Section 3.1 delves into details of TIs and their probabilistic model. This appendix therefore introduces the required mathematical background for TI modelling and also their later aggregation. The stochastic approach to aggregate TIs based on Bayes networks was already previously introduced in Section 2.2 due to its application within STORE's core aggregation component. Further details about the mathematical background as presented in this section can be found in a variety of well established mathematical literature dealing with probability theory such as Prof. Norbert Henze, "Stochastik 1 and 2"[56].

A.1.1 Algebra

Definition 9: A *homomorphism*[19] is a map $f : G \rightarrow H$ from one structure G into another H, e.g. a group or ring, such that the structure's operation is preserved:

$$f(g_1 \circ g_2) = f(g_1) \circ f(g_2) \forall g_1, g_2 \in G \tag{A.1}$$

where the operation on the left-hand side is in G and on the right-hand side in H.

As a result, a homomorphism maps the identity element in G to the identity element in H: $f(e_G) = e_H$.

Note that a homomorphism must preserve the inverse map because $f(g) \circ f(g^{(-1)}) = f(g \circ g^{(-1)}) = f(e_G) = e_H$, so $f(g)^{(-1)} = f(g^{(-1)})$.

A.1.2 Probability Distributions

Before probability distributions can be properly introduced, a set of elemental axioms, concepts and definitions is needed. The following definitions and theorems are part of the axiom system set up by A.N. Kolmogorov[1] (1933).

Definition 10: A non-empty, countable set $\Omega = \{\omega_1, \omega_2, ...\}$ is called an (elemental) *sample space*.

The elements of Ω are called *outcomes*. Each subset A of Ω is called an *event*, each subset consisting of only one element is called an *elementary event*.

Definition 11: A *discrete probability space* is a pair (Ω, P), where Ω is an elemental sample space and P a real valued function defined on the power set $\mathfrak{P}(\Omega)$ of Ω with the following properties:

[1] Andrej Nikolajewitsch Kolmogorov (1903-1987)

$$a) P(A) \geq 0, A \subset \Omega \qquad (Non - Negativity) \tag{A.2}$$

$$b) P(\Omega) = 1 \qquad (Standardised)$$

$$c) P(\sum_{j=1}^{\infty} A_j) = \sum_{j=1}^{\infty} P(A_j) \qquad (\sigma - Additivity)$$

for each sequence $(A_j)_{j \in \mathbb{N}}$ of pairwise disjunct events. P is called the probability measure on Ω, or better on $\mathfrak{P}(\Omega)$. P(A) is also called the probability of the event A.

Definition 12: Let (Ω, P) be a probability space and $A_1, ..., A_n$ events, $n \geq 2$. $A_1, ..., A_n$ are called (stochastically) *independent* if:

$$P(\bigcap_{j \in T} A_j) = \prod_{j \in T} P(A_j), \quad \forall T \subset \{1, 2, ..., n\}, 2 \leq |T| \leq n \tag{A.3}$$

An infinite number of events is called independent if each finite selection of those is independent.

Definition 13: If (Ω, P) is a discrete probability space and Ω' an arbitrary nonempty set, then the mapping $X : \Omega \to \Omega'$ is called a Ω'-valued *random variable*. In the case of $\Omega' = \mathbb{R}^d$, then X is called a d-dimensional random vector, in case of $\Omega' = \mathbb{R}$ short random variable.

Definition and Theorem 1: Through

$$P^X : \begin{cases} \mathfrak{P}(\tilde{\Omega}) \to \mathbb{R} \\ A' \mapsto P^X(A') := P(X^{-1}(A')) \end{cases} \tag{A.4}$$

is a probability measure P^X on a (countable) set $\tilde{\Omega}) := X(\Omega)$ defined. P^X is called distribution of X (under P).

Proof: By verifying the axioms in Equation (A.2), see also [56], p63.

The following probability distributions are defined on probability spaces which is why the latter were defined in adequate length. The formal concept of stochastic independence is instrumental to the design of STORE reputation system instances.

A.1.3 Discrete Probability Distributions

This subsection introduces the main tools for stochastic modelling of TIs. The following set of distributions and their definitions are selected due to their ability to model a core set of TIs that is later also used in the STORE system's evaluation. These distributions proved to be useful in many scientific and industrial disciplines such as risk modelling for insurance and banking.

Each distribution is characterised by its:

- parameter(s) and name

- density function
- expectation value
- variance

A distinction is made between discrete and continuous distributions. In the following, upper case letters denote random variables, e.g. X while upper case letters with an overline denote vectors of random variables, e.g. $\overline{X}$.

In the following paragraphs, let X be a binary random variable.

Binomial Distribution

X is called *binomially distributed* in parameters n and p, $X \sim B(n,p)$, if the density is determined by:

$$P(X=x) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x}, x \in \{1,2,...,n\} \\ 0, otherwise \end{cases} \tag{A.5}$$

where $n \in \mathbb{N}, p \in [0,1]$

The expectation value is defined by $E(X) = np$ and the variance by $var(X) = np(1-p)$.

The binomial distribution is well suited to model the outcome of a repeated binary event e.g. the repeated selection from a population whose members differ in exactly one characteristic.

Bernoulli Distribution

This distribution is related to the Binomial Distribution. If the event which delivers the basis for the definition of X occurs only once, n=1, then $X \sim B(1,p)$ is also called a *Bernoulli distribution*.

Poisson Distribution

X is called *poisson distributed* in the parameter λ, $X \sim P(\lambda)$, if the density is determined by:

$$f(x) = \frac{x^\lambda}{x!} e^{-\lambda}, x = 0,1,2,3,... \tag{A.6}$$

The expectation value is defined by $E(X) = \lambda$ and the variance by $var(X) = \lambda$.

The Poisson Distribution is well suited to model the occurrence of rare events at a given point in time. The nuclear decay of atoms per second or the calls received in a call center per minute are examples for such rare events. Such discrete, rare events in almost all cases repeat themselves over a larger time frame which gives rise to the theory behind *Poisson Processes*. These processes deal with the repeated occurrence of discrete, poisson distributed events.

A.1.4 Continuous Probability Distributions

This subsection deals with continuous distributions.

Normal Distribution

X is called *normal distributed* in the parameters μ and σ, $X \sim N(\mu, \sigma^2)$, if the density is determined by:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} exp(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2) \tag{A.7}$$

The expectation value is defined by $E(X) = \mu$ and the variance by $var(X) = \sigma^2$. In case of $\mu = 0, \sigma = 1$, the distribution becomes the *standard normal distribution*.

The normal distribution plays an important role in stochastic and statistic theory due to its theoretical value e.g. as a limiting distribution of several other continuous and discrete distributions as well as its practical applicability as an "uninformed choice" for a distribution assumption when sampling from a data set that is not necessarily normal distributed.

Lognormal Distribution

The lognormal distribution follows the normal distribution after a transformation. If $Y \sim N(\mu, \sigma^2)$, then $X = exp(Y)$ is *lognormal distributed* in the same parameters $\mu and \sigma$, $X \sim logN(\mu, \sigma^2)$ The density is then determined by:

$$f(x) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma}} \frac{1}{x} exp(-\frac{1}{2}(\frac{ln(x)-\mu}{\sigma^2}), x > 0 \\ 0, x \leq 0 \end{cases} \tag{A.8}$$

The expectation value is defined by $E(X) = e^{\mu - \frac{\sigma^2}{2}}$ and the variance by $var(X) = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1)$.

Random variables that are the product of many independent smaller factors are prone to be lognormal distributed. A random variable modelling a cash flow rate or an environmental risk indicator which are defined by a multitude of independent factors would be examples.

Student's t-Distribution

X is called *student t-distributed* in the parameter μ, $X \sim S(\mu), \mu > 0$, if the density is determined by:

$$f(x) = \frac{\Gamma(\frac{\mu+1}{2})}{\sqrt{\mu\pi}\Gamma(\frac{\mu}{2})}(1 + \frac{x^2}{\mu})^{-(\frac{\mu+1}{2})} \tag{A.9}$$

The expectation value is defined by $E(X) = 0$ for $\mu > 1$ and otherwise undefined. The variance is defined by $var(X) = \frac{\mu}{\mu-2}$ for $\mu > 2$ and otherwise undefined.

The Student's t-Distribution is closely related to the normal distribution. It is symmetric and looks on the first glance similar to a normal distribution with a fatter tail towards the extremes of its support, $-\infty$

and $+\infty$. It is typically employed when e.g. a population is known to behave normally distributed in one property but the standard deviation is unknown. The Student's t-Distribution hereby estimates the standard deviation from the sample data and deals with the inherent uncertainty.

Uniform Distribution

X is called *uniform distributed* on the interval [a, b], a, b $\in \mathbb{R}$, if the density is determined by:

$$f(x) = \begin{cases} \frac{1}{b-a}, x \in [a,b] \\ 0, otherwise \end{cases} \tag{A.10}$$

The expectation value is defined by $E(X) = \frac{a+b}{2}$ and the variance by $var(X) = \frac{(b-a)^2}{12}$.

The uniform distribution is for instance an uninformed choice of a bootstrap distribution assumption for random variables used in a stochastic system with the ability to learn over time.

Exponential Distribution

X is called *exponential distributed* in the parameter $\lambda > 0$, $X \sim E(\lambda)$, if the density is determined by:

$$f(x) = \begin{cases} 0, x < 0 \\ \lambda e^{-\lambda x}, x \geq 0 \end{cases} \tag{A.11}$$

The expectation value is defined by $E(X) = \frac{1}{\lambda}$ and the variance by $var(X) = \frac{1}{\lambda^2}$.

The Exponential Distribution is frequently used to model the inter-arrival time in (stochastic) processes e.g. describing incoming phone calls in a call center or delivery times of goods.

Gamma Distribution

X is called *gamma distributed* in the parameters ν and $\alpha, \nu, \alpha > 0$, $X \sim \Gamma(\nu, \alpha)$, if the density is determined by:

$$f(x) = \begin{cases} 0, x \leq 0 \\ \frac{\alpha^\nu}{\Gamma(\nu)} x^{\nu-1} e^{-\alpha x}, x > 0 \end{cases} \tag{A.12}$$

where $\Gamma(\nu) = \int_0^\infty e^{-t} t^{\nu-1} dt, \nu > 0$.
The expectation value is defined by $E(X) = \frac{\nu}{\alpha}$ and the variance by $var(X) = \frac{\nu}{\alpha^2}$.

The Gamma Distribution with parameter α as an integer is used to model a set of α exponentially distributed random variables with parameter $\frac{1}{\nu}$.

Dirichlet Distribution

The random vector $\overline{Z} = (Z_1, ..., Z_k), k \geq 2$ is called *dirichlet distributed* in the parameter α, $Z \sim D(\alpha)$, if the common density in the inner of the simplex $(z_1, ..., z_k) : \sum_{i=1}^{k} z_i = 1, z_i \geq 0$ is determined by:

$$f(z_1, ..., z_k) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} z_i^{\alpha_i - 1} \tag{A.13}$$

Using the reparametrisation in:

$$\mu_i = \frac{\alpha_i}{\sum_{i=1}^{k} \alpha_i}, i = 1, ..., k or \alpha_i = \mu_i K, i = 1, ..., K := \sum_{i=1}^{k} \alpha_i \tag{A.14}$$

The results in a vector $\overline{\mu} = (\mu_1, ..., \mu_k) = \frac{\overline{\alpha}}{K}$. The expectation value is defined by $E(Z) = \mu$ and the variance by $var(Z_i) = \frac{\mu_i(1-\mu_i)}{K+1}$.

The Dirichlet Distribution is used to model events of k possible outcomes, e.g. feedback, ratings or transactions that can assume k different states [67].

Beta Distribution

The *Beta distribution* is a special case of the Dirichlet distribution for k=2, $Z \sim Beta(\alpha_1, \alpha_2)$. Constraining Z to its first component, $Z = Z_1$, the density is given for $z \in (0, 1)$ by:

$$f(z) = \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} z^{\alpha_1 - 1}(1 - z)^{\alpha_2 - 1} \tag{A.15}$$

The expectation value is defined by $E(Z) = \mu_1$ and the variance by $var(Z_i) = \frac{\mu_1(1-\mu_1)}{K+1}$, using A.14.

Due to the fact, that the Beta is a special case of the Dirichlet Distribution with k=2, it used used to model to model the same types of events but with only binary outcomes, see 2.3.

Erlang-k Distribution

The *Erlang-k Distribution* is defined by $k > 0$, the shape parameter, and $\lambda > 0$, the rate parameter. The density is given for for an erlang-k distributed random variable X by:

$$f(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} \tag{A.16}$$

The expectation value is defined by $E(X) = \frac{k}{\lambda}$ and the variance by $var(X) = \frac{k}{\lambda^2}$.

The Erlang-k distribution is frequently chosen to model average rates in stochastic processes such as an average complaint rate for a product in a call center.

Laplace Distribution

A random variable X is called *laplace distributed* in μ, the location parameter, and $b > 0$, the rate parameter, if the density is determined by:

$$f(x) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}} \tag{A.17}$$

The expectation value is defined by $E(X) = \mu$ and the variance by $var(X) = 2b^2$.

The Laplace Distribution is also often entitled as the double exponential distribution. It is used to model fluctuations over time such as in Brownian Motion or organisational fluctuation situations.

A.1.5 Verification Techniques

Central Limit Theorem

Let $X_1, X_2, X_3, \ldots X_n$ be a sequence of n independent and identically distributed random variables having each finite values of expectation μ and variance $\sigma^2 > 0$. The central limit theorem states that as the sample size n increases, the distribution of the sample average of these random variables approaches the normal distribution with a mean μ and variance σ^2/n irrespective of the shape of the original distribution [56], p139.

Let the sum of n random variables be S_n, given by

$S_n = X1 + ... + X_n$.

Then, defining a new random variable:

$Z_n = \frac{S_n - n\mu}{\sigma\sqrt{n}}$,

the distribution of Z_n converges towards the standard normal distribution N(0,1) as n approaches ∞. This is often written as

$\sqrt{n}\left(\overline{X}_n - \mu\right) \xrightarrow{D} N(0, \sigma^2)$

where

$\overline{X}_n = S_n/n = (X_1 + \cdots + X_n)/n$

is the sample mean (convergence in distribution).

This means: if $\Phi(z)$ is the cumulative distribution function of N(0,1), then for every real number z, we have

$\lim_{n\to\infty} \mathrm{P}(Z_n \leq z) = \Phi(z)$

or,

$\lim_{n\to\infty} \mathrm{P}\left(\frac{\overline{X}_n-\mu}{\sigma/\sqrt{n}} \leq z\right) = \Phi(z)$

Lindeberg Condition

With the same prerequisites as above, only relaxing the requirement of identically distributed random variables, the Central Limit Theorem still holds if the following Lindeberg condition from 1920 holds:

Let X_n be a sequence of independent random variables defined on the same probability space. Assume that X_n has finite expected value μ_n and finite standard deviation σ_n. We define

$s_n^2 = \sum_{i=1}^{n} \sigma_i^2$.

For every $\varepsilon > 0$

$\lim_{n\to\infty} \sum_{i=1}^{n} \mathrm{E}\left(\frac{(X_i-\mu_i)^2}{s_n^2} : |X_i - \mu_i| > \varepsilon s_n\right) = 0$

where $E(U : V > c)$ is $E(U_1 V > c)$, i.e., the expectation of the random variable $U_1 V > c$ whose value is U if $V > c$ and zero otherwise. Then the distribution of the standardized sum Z_n converges towards the standard normal distribution N(0,1).

A.2 List of Trust Indicators

TI name	Abbreviation	Trust class	Subclass	Periodicity	Reference
Relative Market Share	RMS	External	Competition	quarters	Koppelmann21
Ancillary wage costs	AWC	External	Economy	years	Koppelmann21
Average net wages	ANW	External	Economy	years	Koppelmann21
Country Bond (Default) Spread	CBS	External	Economy	daily	Koppelmann21
Economic Growth	EG	External	Economy	quarters	Harting44
ETS on Strike	EOS	External	Economy	irregular	Koppelmann21
Market Growth	MG	External	Economy	quarters	Koppelmann21
Percentage of academics	POA	External	Economy	years	Koppelmann21
Productivity	P	External	Economy	quarters	Koppelmann21
Relative Equity Market Deviation	RMD	External	Economy	daily	Hofmann49
Technological level	TL	External	Economy	years	Koppelmann21
Damage by Natural Disasters	DND	External	Environment	irregular	Eller10
Governmental Subsidy Quote	GSQ	External	Regulations	years	Koppelmann21
Cash Flow Margin	CF	Financial	Balance	quarters	Hofmann49
EBIT Margin	EM	Financial	Balance	quarters	Hofmann49
Equity Ratio	ER	Financial	Balance	quarters	Hofmann49
Equity to Asset Ratio 1,	EAR	Financial	Balance	quarters	(SAP internal)
Economic Value Added	EVA	Financial	Balance	quarters	Hofmann49
Return on Equity	ROE	Financial	Balance	quarters	Hofmann49
Return on Investment	ROI	Financial	Balance	quarters	(SAP internal)
Working capital ratio 1,	WOC	Financial	Balance	quarters	Hofmann49
Beta-Factors	BF	Financial	Stock	daily	Allen6
Earnings per share	EPS	Financial	Stock	quarters	Hofmann49
Price Earnings Ratio	PER	Financial	Stock	daily	Hofmann49
Total Shareholder Return	TSR	Financial	Stock	quarters	Allen6
Contract Penalties	CP	Operational	Legal	irregular	Eller10
Insurance rates	IR	Operational	Legal	irregular	Haller4
Price Stability	PS	Operational	Procurement	irregular	Koppelmann21
Capacity Utilisation	CU	Operational	Production	daily	Large20
Degree of Asset Depreciation	DAD	Operational	Production	quarters	Hofmann49
External Damages	ED	Operational	Production	irregular	Cruz5
Production Flexibility	PF	Operational	Production	irregular	(SAP internal)
Production Risk	PR	Operational	Production	irregular	Hager9
Complaint Rate	CR	Operational	Quality	often	Hofmann49
Waste Rate	WR	Operational	Quality	often	Hofmann49
Customer response time	CRT	Operational	Service	often	Large20
Service Capacity Utilisation	SCU	Operational	Service	daily	Koppelmann21
Delivery Delays	DD	Operational	Shipping	often	Hofmann49
Lead time	LT	Operational	Shipping	often	Chan44
Opportunity Success Rate	OSR	Operational	Shipping	quarters	(SAP internal)
Damage by Systems-Failure	DSF	Operational	Systems	irregular	Hager9
Strategic Funding Systems	SFS	Operational	Systems	quarters	Haller4
System downtime	SD	Operational	Systems	often	Cruz5
Innovation Rate	IR	Organizational	Innovation	quarters	(SAP internal)
Patent Application Rate	PAR	Organizational	Innovation	irregular	Large20
Proprietary Products Sales Rate	PPS	Organizational	Innovation	quarters	Large20
R&D Strategic Funding	RSF	Organizational	Innovation	quarters	Hofmann49
Damage by Management Error	DME	Organizational	Management	irregular	King3
Profit per Head	PPH	Organizational	Personnel	quarters	Hofmann49
Damage by Personnel Errors	DPE	Organizational	Personnel	irregular	Hager9
Employee Fluctuation Rate	EF	Organizational	Personnel	quarters	Haller4
Months of experience	MOE	Organizational	Personnel	quarters	Cruz5
Training hours per FTE	THF	Organizational	Personnel	years	(SAP internal)
Acquisitions of companies	AOC	Organizational	Strategy	years	Haller4
Percent of New Customers	PNC	Organizational	Strategy	years	(SAP internal)

Tab. A.1: List of Trust Indicators

A.3 Reputation Service Screenshot

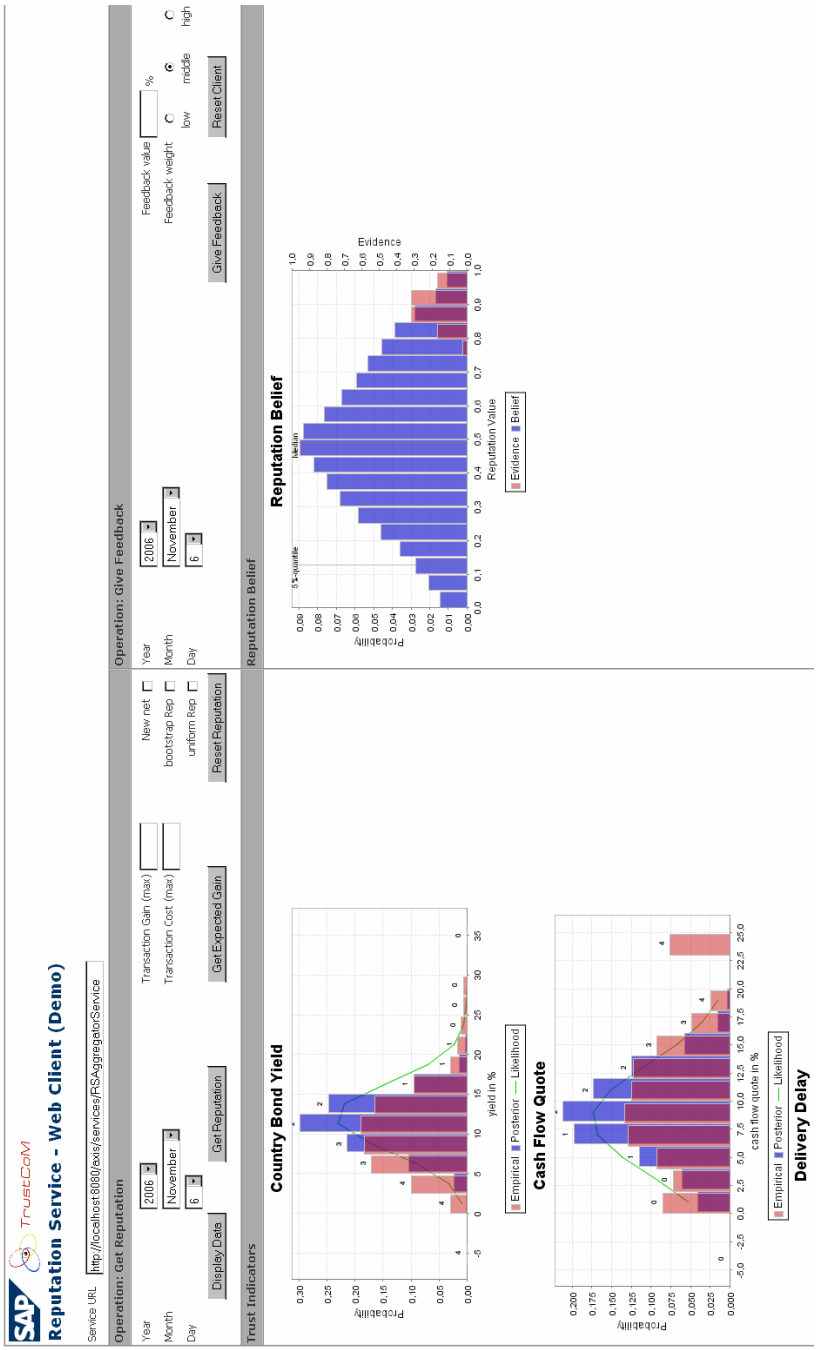

Fig. A.1: Web interface of the Reputation Service Demo

Figure A.3 depicts a screenshot of the first, web service based, STORE research prototype. It is intended to be used by a VOM to query about a potential VO member's reputation. In this prototype, to maintain a lean User Interface (UI), only the generalised reputation node R is included in the graphical output. The TCs are omitted. At the top of the screen, input fields allow for user interaction. A VOM sets a data, for which point in time he intends to query an organisation's reputation. Optionally, he may enter an estimated maximum profit and loss of the transaction, he is going to conduct with a VO member. The checkboxes in the middle of the screen allow to bootstrap a new BN, optionally only with uniform distributions in the TI nodes or also a bootstrapped reputation node according to Subsection 4.2.2. The button "Reset Reputation" executes the checkbox settings. The button "Display Data" would display the raw time series data of each implemented TI below in a chart. The current view depicts the result of a pressed "Get Reputation" button. In the lower left, all distributions, empirical, Likelihood and posterior, of each implemented TI are given in charts. The lower right depicts the distribution of the generalised reputation node R, expectation value and quantiles of interested are marked. The button "Get Expected Gain" evaluates a linear utility function, taking the user input fields "Transaction Gain" and "Transaction Cost" into account. This function linearly scales the reputation distribution from [0, 1] onto the scale ["Transaction Cost", "Transaction Gain"]. The re-scaled expectation value can then be interpreted as the estimated expected gain of this future transaction with this VO member. On the top-right side of the screen, the fields related to the feedback mechanism are shown. Having conducted a transaction with the selected VO Member, the VOM sets the data of the interaction (to provide feedback for exactly this transaction), enters a feedback value in % and an associated measure of confidence (low/middle/high). "Give Feedback" submits the entered feedback. The graphical reputation distribution below is then updated with a visual representation of the triangular feedback distribution according to Subsection 3.2.3.

A.4 Reputation Service Installation Guide

A.4.1 Preface

Introduction

The Reputation Service Demonstration is a prototypical implementation of the trust indicator model presented in section 3.1.1. The implementation was done in Java and consists of a set of three web services and a Java Server Page (JSP) representing a user interface. For the purpose of development, all four services are deployed within the same web container, an Apache Tomcat, on a Windows XP System. As a SOAP engine the Apache Axis 1.2.1 SOAP implementation is used.

Testing

The Reputation Service Demo was tested with a Tomcat 5.0 and J2SDK 1.4.2 as well as with a Tomcat 5.5 and J2SDK 1.5 combination. The latter generally showed better performance and less virtual machine memory errors, particularly caused by the native package of Netica (see A.4.3). However, for compatibility reasons this guide is based on the Tomcat/JDK combination mentioned first. Installation details between the two may vary, therefore see annotations in the following.

Prerequisites

The Reputation Service Demo is based on open source packages. This guide principally only assumes the presence of the project directory `\ReputationService` with the Java sources, archives (JARs) and other

files for the reputation service, all other sources and packages can be downloaded from the web.

A.4.2 Platform and Framework

The setup described here represents a common configuration in Web Service development. That is why only the most important steps are listed here. For further information it is referred to common literature and web resources.

Java

For running Tomcat and the Reputation Service Demo at least a Java 2 Standard Edition Software Development Kit (J2SDK) version 1.4.2 is recommended.

If not present, download from `http://java.sun.com/j2se/1.4.2/download.html` and follow the installation steps.

Set the system environment variable `%JAVA_HOME%` according to the J2SDK directory.

Tomcat

You can get Apache Tomcat at `http://tomcat.apache.org/download-55.cgi`.

Note: The latest version (5.5) requires at least J2SDK 1.5 (or the use of a compatibility package).

Install Tomcat from a Windows executable or a compressed file following the installation instructions.

Set the system environment variable `%CATALINA_HOME%` according to the installation target path.

To verify the installation start Tomcat via the *Tomcat Monitor* icon on the *Windows Start Menu* or by executing `CATALINA_HOME\bin\startup.bat` and open a browser window with the server URL, by default this is `http://localhost:8080`.

For the compilation of JSP sites, Tomcat needs`JAVA_HOME\lib\tools.jar` on its internal classpath. Do this by editing the configuration in the *Tomcat Monitor* or by supplying the additional command line parameter `--Classpath=%JAVA_HOME%\lib\tools.jar`.

Axis

In the demo scenario Axis version 1.2.1 (final) was used. Download Axis from the mirrors given at `http://ws.apache.org/axis/` and unzip it.

Copy the subdirectory `\webapps` to `CATALINA_HOME` and restart Tomcat.

Verify the correct tomcat and axis installation by opening `http://localhost:8080/axis/` in a browser window.

Click on *Validation* (`http://localhost:8080/axis/happyaxis.jsp`) to see, which libraries are missing. Posssibly this will be

- `activation.jar`
- `xmlsec.jar` *(optional)*
- `activation.jar` *(optional)*

Download the missing jar-files or copy them from the project directory `\ReputationService\WEB_INF\lib\*` to `CATALINA_HOME\webapps\axis\WEB-INF\lib`.

Note: Axis requires an XML parser. If you are working on Java 1.4, this step is redundant, since the Java Runtime Environment (JRE) 1.4 comes with the Crimson parser. With Java 1.5 it may be necessary to install a JAXP compatible parser, for example Apache Xerces, downloadable at `http://xml.apache.org/xerces2-j`

To install Xerces the following files should be copied to `CATALINA_HOME\common\lib`

- `xalan.jar`
- `xmlsec.jar`
- `xercesImpl.jar`
- `xml-apis.jar`

Shutdown Tomcat after the correct installation (likewise from the *Tomcat Monitor* or by pressing *Ctrl+C* on the command line).

A.4.3 Project Environment and Development

The adaption to the current system and deployment of the Reputation Service Demo is done from the Eclipse IDE (integrated development environment).

Eclipse

You can download the current version of Eclipse at `http://www.eclipse.org/`. Unzip the installation files to and start `eclipse.exe`.

As a workspace select the parent of the `ReputationService` folder or copy `ReputationService` to your common Eclipse project workspace.

Tomcat PlugIn. For the ease of developing the JSP part of the project and controlling Tomcat directly from Eclipse we use the Sysdeo Tomcat Launcher Plugin for Eclipse, downloadable at `http://www.sysdeo.com/eclipse/tomcatplugin`

Unzip the downloaded file and copy the folder `com.sysdeo.eclipse.tomcat_3.x` to the `eclipse\plugins` directory. Close and restart Eclipse. The three icons for starting, stopping and restarting Tomcat should now be visible on the command bar.

Set the Tomcat Preferences under *Window → Preferences... → Tomcat*, such as the Tomcat Home, which must be the same path as `CATALINA_HOME`, the version number (5.x) and eventually the classpath *(→ JVM Settings)*, which again must include `JAVA_HOME\lib\tools.jar`.

Start Tomcat via the start icon.

ObjectWeb Lomboz (optional). A very powerful Eclipse PlugIn for web development in general is ObjectWeb Lomboz, downloadable at `http://lomboz.objectweb.org`.

This enables amongst others nice graphical displaying of WSDL file structures and hot code debugging in JSP files.

Project Configuration

Now we can add the `ReputationService` folder to our Eclipse workspace.

In Ecplipse create a new Tomcat project (*File → New → Project...*), in the wizard choose *Java → Tomcat Project* and give it the name of the `RepuationService` project folder (which is "ReputationService"). Press *Finish*.

A lot of source files will appear marked red due to missing jar files. Configure the build path in the project properties (accessible via right mouse click on the project root or simply *Alt+Enter*) by adding the missing JARs *Properties → Java Build Path → Libraries → Add JARs...*, as long as there is no more marks. Typically the following JARs will be sufficient

- ReputationService\WEB-INF\lib\lib.axis\axis.jar
- ReputationService\WEB-INF\lib\lib.axis\jaxrpc.jar
- ReputationService\WEB-INF\lib\lib.netica3\NeticaJ.jar
- ReputationService\WEB-INF\lib\lib.math\commons-math.jar
- ReputationService\WEB-INF\lib\lib.jfree\jcommon.jar
- ReputationService\WEB-INF\lib\lib.jfree\jfreechart.jar

Ant file. Apache Ant is a Java and XML based build tool which is supported by Eclipse since version 3.0.

The generation of Web Service related classes with Axis, the compiling and deploying on Tomcat is handled through the Ant script `build.xml` in the project folder.

The Ant script localizes the above mentioned jar files in the project subdirectories (`ReputationService\WEB-INF\lib\*\*`) through its own classpath (called `classpath.ref`), so no changes need to be made here.

For the ease of use, open a new Ant view in Eclipse *(Window → Show View → Ant)*, drag-and-drop the `build.xml` to it and expand the task-tree.

The ant script has to be configured according to the particular setting. Open `build.xml` and customize the values of the following script variables if necessary

- `target.appname` (Name of Axis webapp folder, usually `axis`)
- `target.host` (URI of web container, usually `localhost`)
- `target.port` (Port number of web container, usually `8080`)
- `service.class.dir` (Usually `CATA_HOME\webapps\axis\WEB-INF\classes`)

Note: If Ant should have problems in finding the right compiler (e.g. after upgrading to another Java Version), revise the settings in *Window → Preferences → Ant → Runtime → Global Entries*.

WSDL files. If you changed `target.host` or `target.port`, these changes also have to be reflected in the namespaces of the web services' description files (WSDL).

Open each file under `\ReputationService\wsdl\*.wsdl` and do a find-and-replace of the values.

Source files. A few constants, mainly for file management, are hardcoded within the source files. Adapt these to your needs

- In `RSAggregatorServiceSoapBindingImpl.java`
  - `NET_FILE_NAME` (Filename for net file, default `RSNet.dne`)
  - `NET_FILE_PATH` (Path to save net file, to be set!)
- In `RSDataBaseServiceSoapBindingImpl.java`
  - `DATAFILE_PREFIX` (Prefix of TI data files, default `TiData_`)
  - `DATAFILE_EXT` (Extension of TI data files, default `csv`)
  - `DATAFILE_PATH` (Path to TI data files, to be set!)
- In `RSConstants.java` *(optional)*
  - `DB_START_DATE` (virtual starting date of TI database)
  - `DB_END_DATE` (virtual ending date of TI database)
- As well as the properties of each trust indicator in package `trustIndicators`, whereas those are subject to the overall model.

Deploying

When all settings and adaptions are made, the three Web Services can be deployed to Tomcat. The general proceeding therefore is the following:

1. Execute the *WSDL-2-Java* ant-task, in order to actualize the auto-generated Web Service classes.
2. Compile with the *compile* task, which builds the class-files directly to the web-server location.
3. Start (or restart) Tomcat via clicking the *restart*-icon.
4. Execute the *deploy* ant-task.

You can verify the success of deploying by looking at the list of installed Web Services under `http://localhost:8080/axis/servlet/AxisServlet`

Server-side Configuration

For failure-free running of the Reputation Service Demo a bunch of packages and libraries are required on the web container.

JARs required. The JARs listed here need to be placed in the tomcat shared libraries directory `CATALINA_HOME\shared\lib` in order to be loadable by the Axis Web Services as well as by the JSP-Client.

First move the **Axis** JARs from `CATALINA_HOME\webapps\axis\WEB-INF\lib` to the shared library directory, since they also need to be loadable by the JSP client.

- `activation.jar`
- `axis.jar`
- `axis-ant.jar`
- `commons-discovery.jar`
- `commons-loging.jar`
- `jaxrpc.jar`
- `log4j.jar`
- `mail.jar` *(optional)*
- `saaj.jar`
- `wsdl4j.jar`
- `xmlsec.jar` *(optional)*

The following JARs can also be found in the project directory under `\ReputationService\WEB-INF\lib\*.lib`.

JFreeChart is needed for plotting and displaying graphs.[2] It requires

- `gnujaxp.jar`
- `jcommon.jar`
- `jfreechart.jar`
- `junit.jar`
- `servlet.jar`

[2] JFreeChart is published under GNU Lesser General Public License and sponsored by Object Refinery Ltd., see http://www.jfree.org and http://www.object-refinery.com

Commons Math is the mathematics and statistic package utilized for the representation of the trust indicators.[3] Therefore also copy

- `commons-math.jar`

Netica API. Netica is the package used to represent the Bayesian Network within the aggregator web service.[4] The Netica API unses a Java native interface and requires the following three files, that can be copied from the project directory (`ReputationService\WEB-INF\lib\lib.netica3`).

- `NeticaJ.jar` can be placed in the Axis lib folder (`CATALINA_HOME\webapps\axis\WEB-INF\lib`)
- `Netica.dll` has to be on the system path. Therefore place it in a new directory (for example `C:\DLLs\Netica`) and add this to the path environment variable (or simply place it in `Windows\System32`).
- `NeticaJ.dll` is recommended to be put into the same directory like the above. Additionally you should set a JVM parameter pointing to this library directory like `-Djava.library.path="C:\DLLs\Netica"` when starting Tomcat. In Ecplipse this is done under *Window → Preferences → Tomcat → JVM Settings*.

In order for the system to find the Netica dlls, shut down Tomcat, close Eclipse and restart Windows.

A.4.4 Service Usage and Additional Tools

Now the Reputation Service should be ready for the demonstration. Start Tomcat from Eclipse (or *Tomcat Monitor* or command line).

Web Interface

The web interface, provided by `RSClient.jsp`, was automatically added to the Tomcat web context by Eclipse. Open the page under `http://localhost:8080/ReputationSystem/RSClient.jsp`

The interface is divided into the two basic operations of the ReputationService: *Get Reputation* and *Give Feedback*, each with a set of buttons for the possible actions. The actions referring to *Get Reputation* are:

Display Data. Makes a call only to the database service and displays the available trust relevant data in various time series charts.

Get Reputation. Makes a call to the aggregator service and displays current state of the trust indicator distributions modeled at the supplied date and shows the resulting reputation belief.

Get Expected Gain. Makes a call to the aggregator service and displays current state of the trust indicator distributions modeled at the supplied date and shows the resulting reputation belief as well as the expected gain distribution based on the supplied maximum gain and cost.

Reset Reputation. Makes a call to the aggregator service and resets the reputation belief to either a uniform or a especially bootstrapped distribution.

[3] Math-Commons is a part of the Jakarta Commons sub project and is published under Apache licence, see http://jakarta.apache.org/commons/math

[4] Netica and the Netica API are commercial products and registered trademarks of Norsys Software Corp., Vancouver, Canada. The software is available as a free version with full functionality but limitations in model size (maximum 50 nodes)

The actions referring to *Give Feedback* are:

Give Feedback. Makes a call to the aggregator service and incorporates a feedback value from 0 to 99 with a certain weight, referring to the state of the trust indicators at the supplied date. Also displays the trust indicator distributions.

Reset Client. Returns to the initial page, no form variables are supplied.

Data Input

New data for a trust indicator can be supplied to the database service by placing a flat file in the file location defined by the `DATAFILE_PATH` constant, see A.4.3 (default folder name `tidata`).

In order to be found by the service the filename must carry the respective trust indicator name as assigned in `TiFactory.java`, surrounded by the prefix `TiData_` and extension `.csv`, for example `TiData_CountryBond.csv`.

The file should contain a chronological, semicolon-separated list of date-value pairs, e.g. `31.3.2006;10,5`, followed by a newline-character. The decimal separator may be a comma as well as a point. This kind of data can be extracted and saved with common mathematical and table calculation software, such as Microsoft Excel™ or MathWorks MATLAB.

Netica Application

The aggregator service uses the Netica API to hold a Bayesian network for aggregating the different TI information and feedback.

The network is stored in a file in `\netdata\RSnet.dne` by default, see A.4.3 to change default configuration. If no network file is existent, the service will bootstrap a new aggregation net on the next service call.

This net file can be opened in the Netica application for editing and analysis.[3] The software is downloadable at `http://www.norsys.com/download.html`.

Particularly the incorporation of expert knowledge is done here by defining the preference mapping, cp. 3.1.2. Therefore open any Pi node with double-click, choose *Table* and set the Preferences for each state of your respective TI node.

A second modification can be carried out in case there is no expert knowledge available for a certain trust indicator. Then just delete the link from the respective Pi node to the reputation belief and add a link directly from the TI to the R node. This will correlate trust indicator states with the reputation belief when gaining feedback, see description in 3.2.3.

After editing save the file with *Ctrl+S*, additional display information can be included into the file without problems. The Reputation Service will now consider your changes on the next call when evaluating the reputation belief.

A.5 Software Packages for Bayesian Networks

The following comparison of software packages for graphical models and Bayesian networks is adapted from Kevin Murphy's web page[5] at the department of computer science and statistics, University of British Columbia. Deprecated entries have been removed and links have been updated when necessary. It compiles

[5] `http://www.cs.ubc.ca/~murphyk/Bayes/bnsoft.html`, last updated 31 October 2005

several commercial as well as open source packages from different institutions. The table headers are coded as follows:

Src Source code included? (N=no) If so, what language?

API Application Programming Interface included? (N means the program cannot be integrated into proprietary code, i.e., it must be run as a standalone executable.)

Exec Executable runs on W = Windows (95/98/NT), U = Unix, M = Mac, or - = any machine with a compiler.

Cts are continuous (latent) nodes supported? G = (conditionally) Gaussians nodes supported analytically, Cs = continuous nodes supported by sampling, Cd = continuous nodes supported by discretization, Cx = continuous nodes supported by some unspecified method, D = only discrete nodes supported.

GUI Graphical User Interface included?

Par Learns parameters?

Struc Structure learning supported? CI = means uses conditional independency tests

Util Utility and decision nodes (i.e. decision networks) supported?

Lic License 0 = free (possibly only for academic use), $ = commercial software. (Most packages have free versions which are restricted in various ways, e.g., the model size is limited, or models cannot be saved, or there is no API)

Inference Inference algorithm that is used, JTree = junction tree, VarElim = variable (bucket) elimination, MH = Metropolis Hastings, G = Gibbs sampling, IS = importance sampling, sampling = some other Monte Carlo method, PolyTree = Pearl's algorithm restricted to a graph with no cycles, none = no inference supported (hence the program is only designed for structure learning from completely observed data)

Name	Authors	Src	API	Exec	Cts	GUI	Par	Struc	Util	Lic	Inference
http://www.agendarisk.com											
AgenaRisk	Agena	N	Y	WU	Cx	Y	Y	N	N	$	JTree
http://www.lumina.com											
Analytica	Lumina	N	Y	WM	G	Y	N	N	Y	$	sampling
http://www.cs.duke.edu/~amink/software/banjo											
Banjo	Hartemink	Java	Y	WUM	Cd	N	N	Y	N	0	none
http://www.cs.helsinki.fi/research/fdk/bassist											
Bassist	U. Helsinki	C++	Y	U	G	N	Y	N	N	0	MH
http://cosco.hiit.fi/Projects/NONE/SW											
Bayda	U. Helsinki	Java	N	WUM	G	Y	Y	N	N	0	?
http://www.snn.ru.nl/nijmegen/bayesbuilder.php3											
BayesBuilder	Nijman (U. Nijmegen)	N	Y	W	D	Y	N	N	N	0	?
http://www.bayesia.com											
BayesiaLab	Bayesia Ltd	N	N	-	Cd	Y	Y	Y	N	$	JTree, G
http://b-course.hiit.fi											
B-course	U. Helsinki	N	N	WUM	Cd	Y	Y	Y	N	0	?
http://bnj.sourceforge.net											
BNJ	Hsu (Kansas)	Java	N	-	D	Y	N	Y	N	0	JTree, IS
http://bnt.sourceforge.net											
BNT	Murphy (U.C.Berkeley)	Matlab/C	Y	WUM	G	N	Y	Y	Y	0	Many
http://www.mrc-bsu.cam.ac.uk/bugs											
BUGS	MRC/Imperial College	N	N	WU	Cs	W	Y	N	N	0	Gibbs
http://discover1.mc.vanderbilt.edu/discover/public											
Causal discoverer	Vanderbilt	N	N	W	-	-	N	Y	N	0	-
http://www.cs.ubc.ca/labs/lci/CIspace											
CIspace	Poole et al. (UBC)	Java	N	WU	D	Y	N	N	N	0	VarElim
http://www.math.aau.dk/~jhb/CoCo/cocoinfo.html											
CoCo+Xlisp	Badsberg (U. Aalborg)	C/lisp	Y	U	D	Y	Y	CI	N	0	JTree
http://www.robots.ox.ac.uk/~parg/software.html											
DBNbox	Roberts et al.	Matlab	-	-	Y	N	Y	N	N	0	Various
http://www.math.auc.dk/novo/deal											
Deal	Bottcher et al.	R	-	-	G	Y	Y	Y	N	0	None
http://www.deriveit.com											
DeriveIt	DeriveIt LLC	N	-	-	?	?	Y	Y	?	$	JTree
http://www.bayesware.com											
Discoverer	Bayesware	N	N	WUM	Cd	Y	Y	Y	N	$	?
http://www.noeticsystems.com											
Ergo	Noetic systems	N	Y	WM	D	Y	N	N	N	$	JTree
http://www.staff.ncl.ac.uk/d.j.wilkinson/software/gdagsim											
GDAGsim	Wilkinson (U. Newcastle)	C	Y	WUM	G	N	N	N	N	0	Exact
http://genie.sis.pitt.edu											
Genie	U. Pittsburgh	N	WU	WU	D	W	N	N	Y	0	JTree
http://www.math.ntnu.no/~hrue											
GMRFsim	Rue (U. Trondheim)	C	Y	WUM	G	N	N	N	N	0	MCMC

Tab. A.2: Comparison of software packages for Bayesian networks – continued on next page

Name	Authors	Src	API	Exec	Cts	GUI	Par	Struc	Util	Lic	Inference
http://ssli.ee.washington.edu/~bilmes/gmtk											
GMTk	Bilmes (UW), Zweig (IBM)	N	Y	U	D	N	Y	Y	N	0	JTree
http://www.r-project.org/gR											
gR	Lauritzen et al.	R	-	-	-	-	-	-	-	0	-
http://www.stats.bris.ac.uk/~peter/Grappa											
Grappa	Green (Bristol)	R	-	-	D	N	N	N	N	0	JTree
http://www.hugin.com											
Hugin Expert	Hugin	N	Y	W	G	W	Y	CI	Y	$	JTree
http://www.warnes.net/GregsSoftwareLinks/index_html?Tab=MCMC											
Hydra	Warnes (U.Wash.)	Java	-	-	Cs	Y	Y	N	N	0	MCMC
http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasonng/probabl/0.html											
Ideal	Rockwell	Lisp	Y	WUM	D	Y	N	N	Y	0	JTree
http://www.cs.cmu.edu/~javabayes/Home											
Java Bayes	Cozman (CMU)	Java	Y	WUM	D	Y	N	N	Y	0	VarElim, JTree
http://www.codeas.com/kbaseai.php											
KBaseAI	Codeas	N	Y	WU	D	N	N	N	N	$	VarElim
http://www.cs.huji.ac.il/labs/compbio/LibB											
LibB	Friedman (Hebrew U)	N	Y	W	D	N	Y	Y	N	0	none
http://www.hypergraph.dk											
MIM	HyperGraph Software	N	N	W	G	Y	Y	Y	N	$	JTree
http://research.microsoft.com/adapt/MSBNx											
MSBNx	Microsoft	N	Y	W	D	W	N	N	Y	0	JTree
http://www.norsys.com											
Netica	Norsys	N	WUM	W	G	W	Y	N	Y	$	JTree
http://www.autonlab.org/autonweb/showSoftware/149											
Optimal Reinsertion	Moore, Wong (CMU)	N	N	WU	D	N	Y	Y	N	0	none
http://www.cs.rutgers.edu/~vladimir/pmt/index.html											
PMT	Pavlovic (BU)	Matlab/C	-	-	D	N	Y	N	N	0	special
http://www.intel.com/technology/computing/pnl/index.htm											
PNL	Eruhimov (Intel)	C++	-	-	D	N	Y	Y	N	0	JTree
http://www.cs.ualberta.ca/~jcheng/bnpc.htm											
Power constructor	Cheng (U.Alberta)	N	W	W	D	Y	Y	CI	N	0	?
http://iridia.ulb.ac.be/pulcinella/Welcome.html											
Pulcinella	IRIDIA	Lisp	Y	WUM	D	Y	N	N	N	0	?
http://sourceforge.net/projects/riso											
RISO	Dodier (U.Colorado)	Java	Y	WUM	G	Y	N	N	N	0	PolyTree
http://reasoning.cs.ucla.edu/samiam											
Sam Iam	Darwiche (UCLA)	N	N ?	WU	G	Y	Y	N	Y	0	Recursive conditioning
http://www.phil.cmu.edu/projects/tetrad											
Tetrad	CMU	N	N	WU	G	N	Y	CI	N	0	None
https://sourceforge.net/projects/unbbayes											
UnBBayes	?	Java	-	-	D	Y	N	Y	N	0	JTree
http://vibes.sourceforge.net											
Vibes	Winn & Bishop (U. Cambridge)	Java	Y	WU	Cx	Y	Y	N	N	0	Variational

Tab. A.2: Comparison of software packages for Bayesian networks – continued on next page

Name	Authors	Src	API	Exec	Cts	GUI	Par	Struc	Util	Lic	Inference
http://research.microsoft.com/~dmax/WinMine/tooldoc.htm											
WinMine	Microsoft	N	N	W	Cx	Y	Y	Y	N	0	None
http://www.staff.city.ac.uk/~rgc/webpages/xbpage.html											
XBAIES 2.0	Cowell (City U.)	N	N	W	G	Y	Y	N	Y	0	JTree

Tab. A.2: Comparison of software packages for Bayesian networks – last page

A.6 Simulation Framework Settings

The following configuration files ending with .ini are necessary to configure and run a simulation scenario:

- ScenarioConfig.ini
- QoS.ini
- PreferenceNodes.ini

Following is a detailed description and explanation of how to set up or change these configurations. For legacy reasons, the agent class terminology in the configuration files is not consistent with the classes presented in this thesis. The following Table A.3 presents the mapping of terms.

Agent Class	Term in configuration file
Class 1	Good (agent)
Class 2	GoodAH (agent)
Class 3	GoodCE (agent)
Class 4	Bad (agent)

Tab. A.3: Terminology mapping

A.6.1 ScenarioConfig.ini

This file sets all the necessary variables to run a scenario, configures good and bad Agents defines how the selection and interaction is performed. All lines beginning with "//" are ignored by the Scenario and can be used for comments and notes. Every configuration is done by writing a line with "VariableName=Value", where the order in which the variables are set is not important.

NumberOfScenarios Defines how often this configured scenario should be repeated. For every run, a new set of log-files is created in a /logs subdirectory. To perform an analysis over all executed scenarios, thethe utility class "TestAnalysis.java", that has to be configured by setting variables in the code, can be used. To run the scenario once, set NumberOfScenarios=1.

ScenarioName A name for the scenario. Does not affect the algorithm.

NumberOfBuyers/Sellers The total number of buyers/sellers in this scenario is set by this variable.

NumberOfGoodBuyers/Sellers This setting specifies, how many good buyers/sellers are in this scenario. Together with the above setting "NumberOfBuyers" the number of bad Agents is calculated. Of course the number of good buyers should not be larger than the total number of buyers.

NumberOfGoodAHSellers/NumberOfGoodCESellers The number of the scenario scecialised sellers. See below for definition.

NumberOfSellersForOneBuyer If this number is set bigger than 1 every buyer cooperates with more than one seller each round. This results in more than one pay-off for this buyer. This option is only available for buyers not for sellers.

CostOfRound Every agent pays this amount of money every round just by participating. If you dont want this feature set CostOfRound=0.

BlindRounds A "blind round" is designed to generate regular data observations for the TIs so that historical data is available right from the start. In these rounds no data is added to the reputation system, no QoS is calculated and no transaction performed. It is strongly recommended to make the number of blind rounds at least as high as the number "Rounds as Evidence".

RandomMatching This is a boolean variable. By setting this variable "true" the matching with the help of the reputation system is ignored and an matching between buyer and seller is performed randomly. This generates a benchmark result.

WriteNetToFile This is a boolean variable. If set to "true" for every agent and every round a .dne file with the current Bayes Network (BN) is written to the /netdata subdirectory. This requires a lot of free disk space and eats up performance. Unless needed for debugging, it is strongly recommended to turn this off.

RoundsAsEvidence An integer variable defining how many rounds the BN looks back to collect evidence. Recommended values are 2 for a small interval and 4 for a big.

TI Parameters

For both types of agents "good" and "bad" and every time a TI is simulated the parameters of the distribution that generates the TI values can be defined for each individual round. The structure consists of a part that defines, what kind of agent is regarded, one that specifies the TI and a fixed string "par". If no entry is done for a special point the parameters of the last round are used.

```
"Good"    "CB"  "_par"    = median - stddev     (normal)
"Bad"     "CF"            = median - stddev     (lognormal)
"GoodAH"  "CR"            = shape - rate        (erlang-k)
"GoodCE"  "DD"            = rate                (exponential)
          "SD"            = b - p               (gamma)
          "EF"            = location - scale    (laplace)
```

The Value for each variable is a single string in which the parameters for a round are separated by "-" while the different points of time are separated by ":". For example: To set the expectation value as 5 and the stddev as 1 for the first point in time of the CB distribution of a good agent, and the expectation value as 4 and stddev as 2 for the 3rd point the correct entry is:

```
GoodCB_par=5-1::4-2
```

In this example the parameters for the 2nd point in time are 5-1 too.

Please note: The points in time for which the parameters are set here are NOT the rounds but every time a TI is calculated. If a TI is calculated more than once per round you can enter a different set of parameters for every calculation.

Observations

The number of observations generated, used for the likelihood estimation or as evidence can be configured for each TI individually. Every variable starts with the prefix for the TI it configures.

ObservationPerRound An integer that specifies how often new TI data is available for this TI every round.

ObservationsAsEvidence The evidence is collected every round. This variable defines how many obervations are summarised to one set of evidence. To set how long the evidence includes historical data, use the variable "RoundsAsEvidence".

ObservationsAsLikelihood This variable defines how many observations are used for the likelihood estimation of the distribution. In the current version this variable has no effect as the evidence is entered with the empirical distributions and the net is booted with an uniform distribution as likelihood data.

A.6.2 QoS.ini

The QoS.ini sets several rules that define what kind of TI data will result in what QoS value. The resulting QoS value in that case is the QoS of the best rule that has been evaluated to true. Every rule consists of several attributes, called criteria, that allow for a special TI to be in between an min and a max value. For a rule to be evaluated positively, all the attributes belonging to that rule have to evaluate to true. If a rule doesn't have any attributes, it is always true.

A sample rule may look like that:

```
Rule=Best possible Outcome
Qos=1

Criteria=No Country Risk
TI=CountryBond
Min=0
Max=7

Criteria=Great Cash Flow
TI=CashFlow
Min=5
Max=1000
```

A.6.3 PreferenceNodes.ini

This ini file specifies how the preference nodes CPT tables are set. They consist of a number of values separated by "," that are combined to a float[] while read by the software. The numbers belonging to one float[] don't have to be all in one line to improve readability. The design of the float[] goes like this:

For the first possible combination of states of input nodes all the probability beliefs for all the states of the preference nodes are entered (for example "0,0,0,0,1"), following with the second possible combination of states for all input nodes an so forth.

An example entry may look like this:

```
Node=Pi_DeliveryDelay
0,0,0,0,1
0,0,0,0,1
0,0,0,1,0
0,0,0,1,0
0,0,0,1,0
0,0,1,0,0
0,1,0,0,0
0,1,0,0,0
0,1,0,0,0
1,0,0,0,0
1,0,0,0,0
1,0,0,0,0
```

BIBLIOGRAPHY

[1] *Proceedings of the 1st International Conference on Collaborative Computing: Networking, Applications and Worksharing, San Jose, CA, USA, December 19-21, 2005*. IEEE, 2005.

[2] Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6*, page 6007, Washington, DC, USA, 2000. IEEE Computer Society.

[3] E. Adar and B.A. Huberman. Free riding on gnutella. In *First Monday (Peer-reviewed Journal on the Internet)*, volume 5, October 2000.

[4] Kwabena Adusei-Poku. *Operational Risk management – Implementing a Bayesian Network for Foreign Exchange and Money Market Settlement*. PhD thesis, Faculty of Economics and Business Administration, University of Goettingen, 2005.

[5] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. *Proceedings of the ACM SIGMOD international conference on Management of data*, 2003.

[6] George A. Akerlof. The market for "lemons": Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, 84(3):488–500, 1970.

[7] Steven Allen. *Financial risk management: a practitioner's guide to managing market and credit risk*. Wiley, Chichester [u.a.], 2003.

[8] W. Appelfeller and W. Buchholz. *Supplier Relationship Management: Strategie, Organisation und IT des modernen Beschaffungsmanagements*. Gabler, 2005.

[9] Erich Staudt Arndt. *Kennzahlen und Kennzahlensysteme: Grundlagen zur Entwicklung und Anwendung - Bibliographie deutschsprachiger Verffentlichungen - praxisorientierte Literaturauswertung*. Erich Schmidt, Berlin, 1985.

[10] Ronald Ashri, Sarvapali D. Ramchurn, Jordi Sabater, Michael Luck, and Nicholas R. Jennings. Trust evaluation through relationship analysis. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1005–1011, New York, NY, USA, 2005. ACM Press.

[11] Osman Balci, Randall P. Sadowski, and Richard E. Nance. Models of random machine downtimes for simulation. In *Proceedings of the 1990 Winter Simulation Conference*, 1990.

[12] T. Rev. Bayes. *An Essay Toward Solving a Problem in the Doctrine of Chances*, volume 53. Philos. Trans. R. Soc., London, 1763.

[13] J. Biskup and Y. Karabulut. A hybrid pki model with an application for secure mediation. *16th Annual IFIP WG 11.3 Working Conference on Data and Application Security*, pages 271–282, July 2002.

[14] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *SP '96: Proceedings of the 1996 IEEE Symposium on Security and Privacy*, page 164, Washington, DC, USA, 1996. IEEE Computer Society.

[15] Gary Bolton, Elena Katok, and Axel Ockenfels. How effective are online reputation mechanisms? Technical Report 2002-25, Max Planck Institute of Economics, Strategic Interaction Group, May 2002. available at http://ideas.repec.org/p/esi/discus/2002-25.html.

[16] Padovan Boris, Sackmann Stefan, Eymann Thorsten, and Pippow Ingo. A prototype for an agent-based secure electronic marketplace including reputation tracking mechanisms. Technical Report 0204002, Economics Working Paper Archive at WUSTL, April 2002. available at http://ideas.repec.org/p/wpa/wuwpco/0204002.html.

[17] D. Brickley and R. V. Guha. X.509. IETF RFC 2459, 2004. available at http://www.ietf.org/html.charters/pkix-charter.html.

[18] Clare Brindley. *Supply chain risk*. Ashgate, Aldershot, 2004.

[19] I. N. Bronshtein and K. A. Semendyayev. *Handbook of Mathematics*. Springer-Verlag, New York, 1997.

[20] F. T. S. and Chan. Performance measurement in a supply chain. *Journal Advanced Manufacturing Technology*, 21:534548, 2003.

[21] D. Chaum, C. Crepeau, and I. Damgard. Multiparty unconditionally secure protocols. *Proceedings of the 20th ACM symposium on Theory of computing*, 1998.

[22] P. Closas, C. Fernndez-Prades, and J.A. Fernndez-Rubio. Optimizing the likelihood with sequential monte-carlo methods. *URSI'06 XXI Simposium Nacional de la Unin Cientfica Internacional de Radio, Oviedo, Spain*, 2006.

[23] C. W Cobb. and P. H. Douglas. A theory of production. *American Economic Review*, 18:139–165, 1928.

[24] Marcelo G. Cruz. *Modeling, measuring and hedging operational risk*. Wiley, Chichester [u.a.], 2003.

[25] C. Czernohous, W. Fichtner, D. Veit, and Ch. Weinhardt. Management decision support using long-term market simulation. In *Journal of Information Systems and e-Business Management*, pages 405–423, 2003.

[26] Aswath Damodaran. Measuring company exposure to country risk: Theory and practice. 2003.

[27] W.H. Davidow and M.S. Malone. *The Virtual Corporation*. Harper Business, New York, NY., 1992.

[28] C. Dellarocas. *Building trust on-line: The design of robust reputation mechanisms for online trading communities*. Idea Book Publishing, 2004.

[29] Chrysanthos Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *ACM Conference on Electronic Commerce*, pages 150–157, 2000.

[30] Boris Dragovic, Evangelos Kotsovinos, Steven Hand, and Peter Pietzuch. XenoTrust: Event-based distributed trust management. In *Proceedings of the Second IEEE International Workshop on Trust and Privacy in Digital Business (DEXA Workshop)*, pages 410–414, September 2003.

[31] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J-C. Burgelman. Scenarios for ambient intelligence in 2010. *ISTAG*, 2001.

[32] Paul Kearney (editor). Case study scenarios- WP11 problem definition. *TrustCoM (IST Framework 6 integrated project, grant 001945) public deliverable*, 2004.

[33] R. Eller. *Handbuch operationelle Risiken: aufsichtsrechtliche Anforderungen, Quantifizierung und Management, Praxisbeispiele*. Schffer-Poeschel, Stuttgart, 2002.

[34] Roland Eller. *Handbuch operationelle Risiken: aufsichtsrechtliche Anforderungen, Quantifizierung und Management, Praxisbeispiele*. Schffer-Poeschel, Stuttgart, 2002.

[35] C. Ellison, B. Frantz, B.Lampson, R. Rivest, B. Thomas, and T. Ylonen. Spki certificate theory (rfc2963). The Internet Engineering Task Force (IETF) RFC2693, 1999. available at http://theworld.com/ cme/html/spki.html#1-SPKI/SDSI.

[36] Metropolis et al. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[37] FAA. Faa system safety handbook, chapter 15: Operational risk management. *FAA System Safety Handbook*, December 30 2000. http://www.asy.faa.gov/risk/sshandbook/Chap15_1200.pdf.

[38] R. Falcone and C. Castelfranchi. Social trust: A cognitive approach. In *Trust and Deception in Virtual Societies*, pages 55–99. Kluwer, 2001.

[39] J. Ferber. Addison-Wesley, 1999.

[40] Alberto Fernandes, Evangelos Kotsovinos, Sven Ostring, and Boris Dragovic. Pinocchio: Incentives for honest participation in distributed trust management. In *Proceedings of the 2nd International Conference on Trust Management (iTrust 2004)*, pages 63–77, Oxford, UK, March 2004. Also published in Springer-Verlag Lecture Notes in Computer Science (LNCS), Volume 2995, pp. 63-77.

[41] E. Friedman and P. Resnick. The Social Cost of Cheap Pseudonyms. In *Journal of Economics and Management Strategy*, volume 10, pages 173–199, May 21 2001.

[42] Karen K. Fullam, Tomas B. Klos, Guillaume Muller, Jordi Sabater, Andreas Schlosser, Zvi Topol, K. Suzanne Barber, Jeffrey S. Rosenschein, Laurent Vercouter, and Marco Voss. A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 512–518, New York, NY, USA, 2005. ACM.

[43] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–14, 1962.

[44] Diego Gambetta. *Can We Trust Trust?*, pages 213–237. Basil Blackwell, 1988. Reprinted in electronic edition from Department of Sociology, University of Oxford, chapter 13, pp. 213-237.

[45] Anurag Garg, Alberto Montresor, and Roberto Battiti. Reputation lending for virtual communities. In *ICDEW '06: Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06)*, page 22, Washington, DC, USA, 2006. IEEE Computer Society.

[46] R. Grenier and G. Metes. *Going Virtual: Moving Your Organization into the 21st Century*. Prentice-Hall, Englewood Cliffs, NJ., 1995.

[47] Peter Hager. *Corporate risk management: cash flow at risk and value at risk*. Bankakad.-Verl., Frankfurt am Main, 2004.

[48] S.R. Hakim and D. Shimko. The impact of firm's characteristic on junk-bond default. *Journal of Financial and Strategic Decisions*, 8(2):47–55, 1995.

[49] Jochen Haller. A stochastic approach for trust management. In *International Workshop on Security and Trust in Decentralized/Distributed Data Structures (STD3S)*, 2006.

[50] Jochen Haller. Store stochastic reputation service for virtual organizations. In *Joint iTrust and PST Conferences on Privacy, Trust Management and Security*, June 2008.

[51] Jochen Haller, Philip Robinson, and Yuecel Karabulut. Security controls in collaborative business processes. In *6th IFIP Working Conference on VIRTUAL ENTERPRISES (PRO-VE'05)*, 2005.

[52] Jochen Haller and Christian Wolter. Trust Indicator Integration into SLAs for Virtual Organisations. In *Proceedings of eChallenges 2007 Conference*, 2007.

[53] Roger M. Hayne. Modeling parameter uncertainty in cash flow projections. 1999.

[54] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *J. Mach. Learn. Res.*, 1:49–75, 2001.

[55] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Myers Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

[56] Norbert Henze. *Stochastik fr Einsteiger*. Number 7. Vieweg+Teubner, 2008.

[57] Jeff Herbert. Introducing security to the small business enterprise. *GIAC*, 2003.

[58] Ludger Hinners-Tobraegel. Eigenschaften zusammengesetzter zufallsvariabler analytische ableitungen und monte-carlo-simulationen. *Gesellschaft fuer Informatik in der Land-, Forst- und Ernaehrungswirtschaft, Jahrestagung*, 2002.

[59] Ludger Hinners-Tobraegel. Zur analyse der berlebensfhigkeit von unternehmen methodisch-theoretische grundlagen und simulationsergebnisse. 2002.

[60] Ines Hofmann, Heinz Leitsmueller, and Ruth Naderer. *Unternehmenskennzahlen - Werkzeuge fuer professionelle Betriebsratsarbeit*. Wien, 2001.

[61] Robert W. Irving. The man-exchange stable marriage problem. 2005.

[62] R. Ismail and A. Jøsang. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, 2002.

[63] Finn V. Jensen. *Bayesian networks and decision graphs*. Springer, Tokyo, 2001.

[64] Audun Jøsang, Shane Hird, and Eric Faccer. Simulating the effect of reputation systems on e-markets. In *iTrust*, pages 179–194, 2003.

[65] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, August 2004.

[66] A. Jsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.

[67] Audun Jsang and Jochen Haller. Dirichlet reputation systems. In *ARES*, pages 112–119. IEEE Computer Society, 2007.

[68] Yuecel Karabulut. Towards a next-generation trust management infrastructure for open computing systems. *SPPC: Workshop on Security and Privacy in Pervasive Computing*, 2004.

[69] Yuecel Karabulut, Florian Kerschbaum, Fabio Massacci, Philip Robinson, and Artsiom Yautsiukhin. Security and trust in it business outsourcing: a manifesto. *2nd International Workshop on Security and Trust Management*, 2006.

[70] Jack L. King. *Operational risk: measurement and modelling*. Wiley, Chichester [u.a.], 2001.

[71] Udo Koppelmann. *Beschaffungsmarketing*. Springer, Berlin [u.a.], 1995.

[72] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. pages 415–448, 1990.

[73] Frank Lehrbass. A simple approach to country risk, 2000.

[74] H.C. Lim, C.B. Tab, L.G. Goh, and S.L. Ling. Why do patients complain? a primary health care study. *Singapore Med. Journal*, 39(9):390–395, 1998.

[75] Niklas Luhmann. *Familiarity, Confidence, Trust: Problems and Alternatives*, pages 94–107. Basil Blackwell, 1988. Reprinted in electronic edition from Department of Sociology, University of Oxford, chapter 6, pp. 94-107.

[76] Mass Soldan Lund and Fredrik Vraalsen. Analysing trust, security and legal issues using coras. In *Proceedings of the 3rd International Conference on Trust Management (iTrust)*, 2005.

[77] Neil M and Fenton NE. Improved methods for building large-scale bayesian network. In *The Third Bayesian Modeling Applications Workshop, Uncertainty in Artificial Intelligence (UAI) 2005, Edinburgh University, 26 July*. UAI, 2005.

[78] Anders L. Madsen and Finn V. Jensen. Lazy propagation: a junction tree inference algorithm based on lazy evaluation. *Artif. Intell.*, 113(1-2):203–245, 1999.

[79] M.A.Patton and A.Jsang. Technologies for trust in e-commerce. *In the proceedings of the IFIP working conference on E-Commerce, Salzburg, Austria*, June 2001.

[80] D.B. Miller, E.K. Clemons, and M.C. Row. *Information technology and the global virtual corporation*. Harvard Business School Press, Boston, MA, 1993.

[81] S. Moss and P. Davidsson. Springer, Heidelberg, 2001.

[82] A. Mowshowitz. Virtual organization: A vision of management in the information age. *The Information Society*, pages 267 – 288, 1994.

[83] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation. *35th Annual Hawaii International Conference on System Sciences (HICSS-35)*, 7:188, 2002.

[84] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. Notions of reputation in multi-agents systems: a review. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 280–287, New York, NY, USA, 2002. ACM.

[85] M. Neil and L. Fenton. Building large-scale bayesian networks, 1999.

[86] Martin Neil, Norman Fenton, and Lars Nielson. Building large-scale bayesian networks. *Knowl. Eng. Rev.*, 15(3):257–284, 2000.

[87] K. Neumann. *Dynamische Optimierung, Lagerhaltung, Simulation, Warteschlangen*. Springer, 1977.

[88] P. Nurmi. Bayesian game theory in practice: A framework for online reputation systems. Technical report c-2005-10, University of Helsinki, Department of Computer Science, 2005.

[89] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[90] Judea Pearl. Bayesian networks. Technical Report 980002, 31, 1998.

[91] S. James Press. *Subjective and objective Bayesian statistics: principles, models, and applications*, volume 2. ed. Wiley-Interscience, 2003.

[92] L. Rasmusson and S. Janssen. Simulated social control for secure internet commerce. In Catherine Meadows, editor, *Proceedings of the 1996 New Security Paradigms Workshop*. ACM, 1996.

[93] S. Reece, A. Rogers, S. Roberts, and N. R. Jennings. Rumours and reputation: Evaluating multi-dimensional trust within a decentralised reputation system. In *The Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-07)*, 2007.

[94] Kevin Regan, Robin Cohen, and Pascal Poupart. The Advisor-POMDP: A principled approach to trust through reputation in electronic markets. In *Proceedings of Privacy, Security and Trust (PST05)*, 2005.

[95] P. Resnick and R Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. Technical report, University of Michigan, 2001.

[96] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems: Facilitating trust in internet interactions. *Communications of the ACM*, 43(12):45–48, 2006.

[97] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on ebay: A controlled experiment, 2003.

[98] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, 2000.

[99] Robert Rider. The evolution of cooperation : Axelrod, robert, (basic books, inc., 1984) pp. 256, $17.95. *Journal of Economic Behavior & Organization*, 5(3-4):406–409, 1984.

[100] Rafael Rob and Arthur Fishman. Is bigger better? customer base expansion through word-of-mouth reputation. *Journal of Political Economy*, 113(5):1146–1175, October 2005. available at http://ideas.repec.org/a/ucp/jpolec/v113y2005i5p1146-1175.html.

[101] Philip Robinson, Yücel Karabulut, and Jochen Haller. Dynamic virtual organization management for service oriented enterprise applications. In *CollaborateCom* [1].

[102] Frank Romeike. *Modernes Risikomanagement: die Markt-, Kredit- und operationellen Risiken zukunftsorientiert steuern*. Wiley-VCH, Weinheim, 2005.

[103] Sini Ruohomaa and Lea Kutvonen. Trust management survey. In *Trust Management: Third International Conference, iTrust 2005, Paris, France, May 23-26, 2005. Proceedings Trust Management: Third International Conference, iTrust 2005, Paris, France, May 23-26, 2005. Proceedings*, volume 3477, pages 77–92. Springer, Oxford, UK, 2005.

[104] Jordi Sabater and Carles Sierra. Regret: reputation in gregarious societies. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 194–195, New York, NY, USA, 2001. ACM Press.

[105] Andreas Schlosser, Marco Voss, and Lars Brückner. On the simulation of global reputation systems. *Journal of Artificial Societies and Social Simulation*, 9(1):4, 2005.

[106] Wolfgang Schultze. *Methoden der Unternehmensbewertung: Gemeinsamkeiten, Unterschiede, Perspektiven*. IDW-Verl., Düsseldorf, 2003.

[107] Sandip Sen and Neelima Sajja. Robustness of reputation-based trust: boolean case. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 288–293, New York, NY, USA, 2002. ACM Press.

[108] J. S. Sichman, R. Conte, and N. Gilbert. Springer, Heidelberg, 1998.

[109] S. A. Sisson, Y. Fan, and Mark M. Tanaka. Sequential monte carlo without likelihoods. *PNAS*, 104(6):1760–1765, February 2007.

[110] T.J. Strader, F. Lin, and M.J. Shaw. Information structure for electronic virtual organization management. *Decision Support Systems*, pages 75–94, 1998.

[111] D. Golby T. Dimitrakos and P. Kearney. Towards a trust and contract management framework for dynamic virtual organisations. In *eChallenges Conference e-2004*. IOS press, November 2004.

[112] Yao-Hua Tan. A trust matrix model for electronic commerce. In *Trust Management, First International Conference, iTrust 2003*, volume 2692, pages 33–45, Heraklion, Crete, Greece, 2002. Springer.

[113] W. T. Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

[114] D. Teitelbaum and R. Axtell. *Firm Size Dynamics of Industries: Stochastic Growth Processes, Large Fluctuations, and the Population of Firms*. 2005.

[115] Volker Tresp. Dirichlet processes and nonparametric bayesian modelling, 2006.

[116] TrustCoM. Trustcom - a trust and contract management framework enabling secure collaborative business processing in on-demand created, self-managed, scalable, and highly dynamic virtual organisations. European Union, 6th Framework Information Society, 2004. available at http://www.eu-trustcom.com/.

[117] Hal R. Varian. *Microeconomic Analysis*. W. W. Norton & Company, February 1992.

[118] D. Veit, W. Fichtner, and M. Ragwitz. *Multi-Agenten Systeme als Methode zur Simulation von Entscheidungsprozessen in der Energiewirtschaft*. Shaker, Aachen, Germany, 2004.

[119] Lea Viljanen. Towards an ontology of trust. In *Proceedings of the 2nd International Conference on Trust, Privacy and Security in Digital Business*, volume 3592, page 175. Springer, Copenhagen, 2005.

[120] Yao Wang and Julita Vassileva. Bayesian network-based trust model. In *WI 2003: Proceedings of the IEEE/WIC International Conference on Web Intelligence*, page 372, Washington, DC, USA, 2003. IEEE Computer Society.

[121] Ingo Weber, Jochen Haller, and Jutta A. Mülle. Automated derivation of executable business processes from choreographies in virtual organizations. In *Third GI-Workshop XML4BPM XML Integration and Transformation for Business Process Management*, 2006.

[122] M. West. Mixture models, monte carlo, bayesian updating and dynamic models. *Computing Science and Statistics*, 24:325–333, 1993.

[123] Edward J. Williams. Downtime data - its collection, anaysis, and importance. 1994.

[124] Till J. Winkler, Jochen Haller, Henner Gimpel, and Christof Weinhardt. Trust indicator modeling for a reputation service in virtual organisations. In *'The 15th European Conference on Information Systems (ECIS)*, 2007.

[125] Till J. Winkler, Jochen Haller, Henner Gimpel, and Christof Weinhardt. Trust indicator modeling for a reputation service in virtual organisations. In *to appear: Journal of IS Security (JISSec)*, 2007.

[126] G. Wohland. *Das widerständige Nest - Anforderungen an die Umgebung von Projektteams mit höchster Leistung*. Detecon Management Report, Heft 3/2005, 2005.

[127] M.J.J. Wolters and M.R. Hoogeweegen. Management support for globally operating virtual organizations: The case of klm distribution. *Proceedings of the 32nd Hawaii International Conference on System Sciences (HICSS-32)*, pages 5–8, January 1999.

[128] Li Xiong and Ling Liu. A reputation-based trust model for peer-to-peer ecommerce communities. In *Proceedings of the IEEE International Conference on E-Commerce Technology (CEC'03)*, page 275, 2003.

[129] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. Collaborative reputation mechanisms in electronic marketplaces. In *Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences (HICSS-32)*, 1999.