# Abstract

Researchers from the social sciences and economics consider trust a requirement for successful cooperation between people. It helps to judge the risk in situations, in which a person has the choice to rely on another one. In the future, technical systems will face similar situations. Assume for example that, at a large logistics centre, a robot should reload goods of a ship in cooperation. In the beginning, it must find the right partner out of a set of diverse other robots. To do this selection efficiently without exaggerated security mechanisms, the robot needs trust. Here I consider trust a mechanism, which estimates the certainty of the outcome of the partner's actions.

This dissertation formalises trust between technical systems to set the theoretical foundation for the above idea. It reviews the socio-scientific and technical literature and identifies generic requirements for the mechanism trust. Based on the requirements and further considerations, it presents a conceptual, implementation-independent framework. This new framework, called the Enfident Model, incorporates various facets of trust in form of submodels. Amongst others, it regards the temporal development of cooperation, the dependency on the task and bargaining, time-varying behaviour of the cooperation partner, learning from experiences, logical constraints of the present situation, and transfer learning to handle unknown situations. With these manifold features described on a conceptual level, the Enfident Model captures existing trust procedures and is suitable for designing new ones. The theoretical part is complemented with algorithms for prototyping trust in individual applications. These algorithms use statistical relational learning to combine logic, learning, clustering and statistics for trust development. They work on a relational dynamic Bayesian network.

Since trust is a social phenomenon, the evaluation features a virtual society of vehicles. These systems cooperate by exchanging information in a vehicular network. They use a trust algorithm to distinguish correct from incorrect information. The simulation shows that the identified trust requirements and the Enfident Model lead to intuitive and consistent results.

# 1 Introduction

In a future with many self-organising systems, socio-scientific issues also apply to the society of those machines. Imagine, for example, a future scenario of robots at a large construction site. They have different shapes and abilities as they have been optimised for different purposes, like moving big and heavy items, or cutting and screwing. Some of them have worked together before; others do not know each other, because they are new or belong to different companies.

In this scenario, various complex tasks can only be executed jointly by a group of robots. Imagine a robot has got the job to carry out such a task. It looks for partners, asks them whether they would be willing to do the job, and finally performs the task with their support. Selecting the right cooperation partners is important for an optimal outcome: The partner could have insufficient abilities, be partly defect, or be manipulated to sabotage the task. Thus the organising robot should select those partners, with which it expects to gain the best outcome. This is where trust comes in.

## 1.1 Problem Statement

The scenario above is an example for the problem this dissertation addresses. The general setting consists of a system that wants to cooperate with another system or a group of systems. Here I understand cooperation as any form of relying on the action of another party. That setting is related to various subjects like reputation, identification of the partner, individual trust development, decision making, reciprocity and information security (see Figure 1.1 on page 3). This dissertation picks out just one. It focuses on the single problem: *How can a system that wants to cooperate with another sys-*

*tem or a group of systems predict the cooperation outcome?* This prediction should have the form of beliefs in or likelihoods for all possible cooperation outcomes.

The problem can also be considered from another point of view: If a system can predict cooperation outcomes, it has a certain model of the other's manifest behaviour. It cannot look into the other system to see how that system really works. But it can obtain a limited idea of how the other system works just from observing its behaviour over several interactions. This idea is a model of the other's manifest behaviour regarding cooperation. So the problem treated in this dissertation can also be formulated as: *How can a system learn a model of other systems' cooperation-related behaviour?*

I call a mechanism, which can learn this, *trust between technical systems*. The term trust has different meanings in different fields. To address this fact, the next chapter introduces the views of some researchers in social sciences, cryptology and the field of multi-agent systems. It relates them to the problem described above to clarify why I use the term trust in this dissertation. Finally it defines some trust-related terms for the present work. Chapter 4 summarises the state of the art for technical trust mechanisms. The contribution of this dissertation beyond the state of the art is compiled in Section 1.3.

More specifically, this dissertation does not try to simply solve the described problem with a certain algorithm for a specific application. Instead it collects requirements for a trust mechanism in general and derives a conceptual trust model from them. To realise and evaluate this model, an exemplary algorithm is presented. More implementations of the model and optimisations are subject to future research.

In the remainder of this section, I further detail the problem and delimit it from selected other problems that the reader may possibly think of. For this, Figure 1.1 gives some orientation. The term cooperation is interpreted very widely in this document. It includes delegation and all sorts of relying on another party. Consider, for example, a driver that is overtaking another car on a highway. The situation seems free of risk as no third car is around. But still, each of the drivers relies on the other one not to hit the own car (for whatever strange reasons). This situation features a form of loose re-

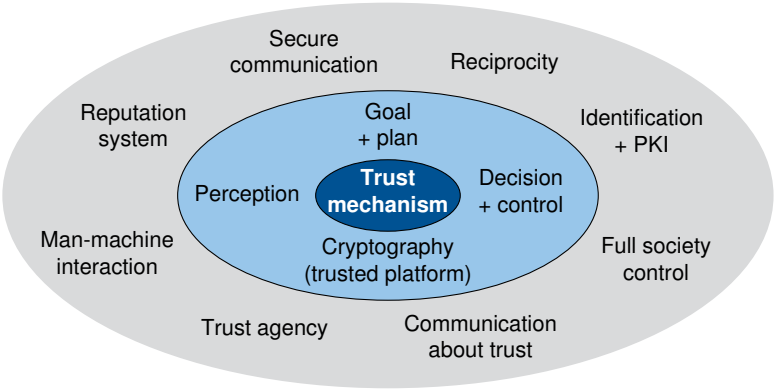


**Figure 1.1:** This figure shows some mechanisms the reader may think of when talking about trust. The blue ellipse contains modules that work on the individual level. Those in the grey part are used for the interaction with systems: the society level. This dissertation only treats the trust mechanism marked in dark blue.

liance without any explicit agreement. In this dissertation, I still consider it an implicit form of cooperation, as it constitutes a trust situation.

Furthermore the systems here should cooperate without human support. Especially they should develop trust on their own. This is in contrast to systems that use humans as trust sources like classical online reputation systems. That points to an important pre-requisite: In this thesis, a trusting system must be able to assess all facets of a cooperation outcome. Only then, it can learn the cooperation-related behaviour of others on its own.

Related trust methods often include mechanisms for decision making, reputation building, reciprocity enforcement as well as cryptographic data and platform security. I focus on the trust development in the individual and omit society-level features like cryptographic network protocols or reputation building. Moreover I consider decision making and also reciprocity to be different from trust development (see Chapters 2 and 3).

So I propose a mechanism, which just learns a model of the other's behaviour. All the tools mentioned above are related to trust and important for a trusting society. Figure 1.1 depicts this. But they are different from

a trust mechanism in the strict sense that is proposed in this dissertation. Moreover the dissertation focuses on machine-machine interaction without human intervention. Every time, when cooperating and trusting systems or agents are mentioned, I refer to technical systems, except if interpersonal trust is considered explicitly.

What comes very close to a trust mechanism is a sensor model. Such a model describes how a sensor transforms the observed physical quantity in an output signal. So it reflects the behaviour of a sensor. A trust mechanism goes beyond this. It learns behavioural models for many other systems, not just one sensor, and for many tasks, not just the single task of obtaining a certain physical quantity. In addition, these other systems are unknown in advance and their basic way of functioning may vary. Still the trust mechanism should provide accurate expectations, even if only few experiences have been made with the other systems before. Thus the trust mechanism must be able to learn various behavioural models; it must be generic. And it should involve transfer learning to quickly adapt to new situations.

The next section introduces various scenarios in which a technical form of trust is useful. The scenarios show that the present work has relevance for the research on cognitive systems, multi-agent systems, sensor networks, vehicular networks and – to some extent – on cryptology; it features techniques from the field of statistical relational learning.

## 1.2 Motivation and Applications

Trust is only a minor subject in the development of today's technical systems. In contrast to this, interpersonal trust is considered important for personal relationships as well as business organisations (see Section 3.3 and, e.g., Gennerich, 2000, pp. 10–12 for an overview). It improves communication and cooperation, and it is considered a pre-requisite of efficient work flows in groups. If it is so important for people, why is it used only rarely in technical systems? The main reason might be that trust is especially necessary for cooperation between self-organising agents. Strictly controlled work flows, as they are typical today for machine-to-machine interaction, make

trust needless. But the proposed idea is important for systems that cooperate in a self-organised way. Such systems will need a trust mechanism to handle the uncertainty when relying on other systems. As a consequence, the reader should venture a glimpse into the future to find application scenarios for trust between cooperating systems.

I use the following exemplary scenarios throughout this dissertation. The first is the scenario of a construction site as described in the previous section. It is similar to the scenario of a large logistics centre with various kinds of robots that cooperate to reload goods from a ship. In both scenarios, the cooperation helps to extend the physical capabilities or to perform tasks more efficiently. In the third example, future cognitive vehicles are driving around while perceiving their environment. To extend their perception range, they exchange all sorts of information, which some vehicles have perceived before. With this form of cooperation, they can efficiently maintain a model of their surrounding world (like a map or a model of the traffic situation) and advise the driver (e.g., where to go or what to give attention to). The fourth scenario features virtual agents at a virtual market place, which trade with each other. So they cooperate as substitutes of persons. These scenarios should give the reader the feeling that trust is helpful for future self-organising systems.

In general, trust supports the following reasoning tasks that appear when cooperating:

1. Select a cooperation partner from several possible ones;
2. Decide whether to cooperate or not if there is a choice not to cooperate at all;
3. Know about the weaknesses of a certain act of cooperation and take their consequences into account;
4. Decide about the correctness of received information;
5. Decide whether the received information about a certain subject is sufficient; and if not,

6. Decide whom to ask for a further opinion about the subject (which is related to Item 1); and finally,

7. Decide whether to accept a cooperation request of another party (which is related to Item 2). So trust is usually needed by both, the one that asks for cooperation and the other one that is asked.

In summary, a self-organising cooperating system needs trust to decide on "how, when, and who to interact with" (Ramchurn et al., 2004, p. 3).

The reader can find many scenarios, in which future technical systems could perform the above reasoning tasks. To support this, I give an overview of the various forms of cooperation, which can be expected in the future (based on Hirche, 2010). It was proposed in CoTeSys, a cluster of excellence of the Deutsche Forschungsgemeinschaft (German Research Foundation), which investigates cognitive systems.

- Two systems interact solely through the environment during the cooperation.

- Two systems share single components and couple one another via information exchange
  - to extend the perception range (joint perception),
  - to extend the physical capabilities (joint manipulation),
  - to increase the learning performance (joint learning), and
  - to find good and efficient strategies for the task execution (joint planning and decision making).

  Figure 1.2 illustrates how two cognitive systems can share various components directly.

Both main forms of cooperation can also be mixed. Trust is helpful in all cases.

With this schema of cooperation forms, the reader might get an idea of the various applications we can expect of future self-organising systems. The previous list of reasoning tasks shows that a trust mechanism can strongly

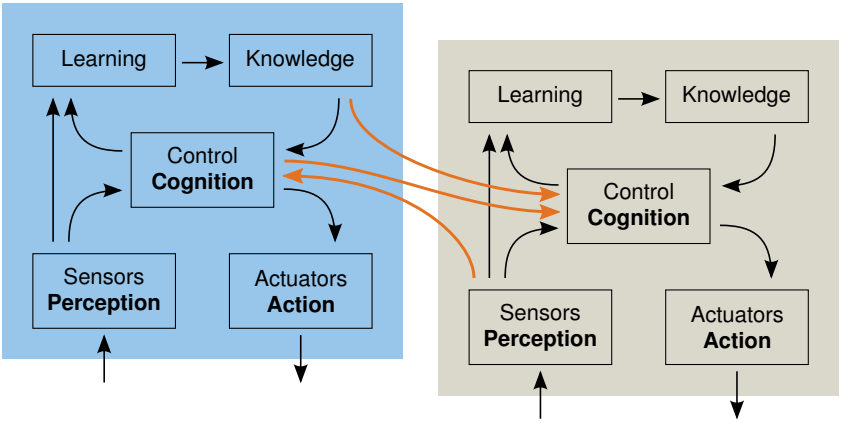


**Figure 1.2:** Examples of how two cognitive systems can share their components (based on Hirche, 2010). The black lines indicate data flows between the components in one system. The orange lines refer to data flows, which are realised by communication between two different systems.

support the reasoning in these applications. So there is a wide range of use cases trust can be applied to. But is trust really necessary or could it be substituted with better planning and control in the scenarios? Full control over complex situations with several interested parties is difficult and, thus, expensive. Imagine, for example, a large harbour in the future. The robots there belong to different parties, have various ages and come from several manufacturers. So full control is difficult here. Avoiding strict global control is the exact idea behind self-organising systems. Thus trust enables those systems to cooperate efficiently without expensive procedures for security enforcement. This concern is similar to that of Gerck (2002), who recommends trust for the Internet because of its self-organising nature. For him, using trust instead of full surveillance has the advantages of a simpler and more modular system design as well as lower costs.

Above I used the notion of a cognitive system. This kind of system has the ability to trust, because it can perceive and understand its environment in order to judge past acts of cooperation and to learn from them. And this kind of system has a need for trust, because it should engage in cooperation and

reason about cooperation. Therefore cognitive systems are widely used in this dissertation, but the application of trust is not limited to them. This term is defined in CoTeSys as follows:

> "*Cognitive technical systems (CTS)* are information processing systems equipped with artificial sensors and actuators, integrated and embedded into physical systems, and acting in a physical world. They differ from other technical systems as they perform *cognitive control* and have *cognitive capabilities*. *Cognitive control* orchestrates reflexive and habitual behavior in accord with longterm intentions. *Cognitive capabilities* such as perception, reasoning, learning, and planning turn technical systems into systems that '*know what they are doing*'." (Buss et al., 2007, p. 25)

## 1.3 Contribution of This Dissertation

This dissertation has the objective to improve the understanding and modelling of trust between cooperating technical systems. To achieve this, it contributes the following to a theory of technical trust.

It discusses the term and mechanism "trust" across disciplines and introduces research on interpersonal and technical trust to compare various views. In contrast to the state of the art (e.g. Castelfranchi and Falcone, 2010; Engler, 2007; Kassebaum, 2004), this dissertation presents *interpersonal trust as an input-output system*. This new view makes it easier to relate trust between persons and between machines with each other. In addition, the presented interdisciplinary discussion is deeper than the state of the art. This leads to a different understanding of technical trust, especially regarding the following questions: What notions of trust can be distinguished (Section 2.5)? How does trust differ from related mechanisms (Chapters 1 and 2)? What influences trust development (Sections 3.2 and 6.2)? How do interpersonal trust and inter-machine trust differ from one another (Sections 3.4 and 10.2.5)? This work results in clear, well-founded technical con-

cepts for different notions of inter-machine trust. It is necessary, because the present state of the art lacks a sufficient theoretical framework for the trust model presented in this document.

The interdisciplinary research together with an analysis of future trust scenarios leads to a formalisation of trust between technical systems. This formalisation is the core contribution of this dissertation. It consists of *general application-independent requirements* on a trust algorithm and a *conceptual implementation-independent model of trust*. The requirements are postulated together with a review of the technical literature in Chapter 4. Formal requirements for a trust mechanism are unique in the literature. While some authors (e.g. Ramchurn et al., 2004) review the literature on trust, they do not derive requirements from it. Furthermore the new conceptual model of trust describes various aspects of trust development and can be understood as a meta-model to create new application-specific trust algorithms. It is presented in Chapter 6 and called the *Enfident Model*. The following list details its main features with a focus on those that are rarely found in other trust models.

- The Enfident Model *evaluates a trust situation comprehensively*. It explicitly names three aspects: the cooperation partner(s), the cooperation agreement and the task to fulfil. It combines them as entity classes in a relational sub-model; each of the entity classes groups several attributes of the trust situation. Present trust models consider the attributes of one or two of those entity classes only, as Section 4.2 points out.

- This relational sub-model can reunite two lines of research on technical trust, which are detailed in Section 4.2. Today, most trust algorithms rate previous cooperation outcomes and derive trust from these ratings. In contrast, the socio-cognitive trust models derive trust from beliefs about the cooperation partner in the given trust situation, basing their theory on belief-desire-intention agents. *These beliefs can be located in the Enfident Model in the same way as the cooperation outcomes and contextual information.*

- Section 4.4.1 shows that some trust algorithms base their outcome prediction on *past experiences*, while others use *logical constraints of the present situation*. The Enfident Model addresses both information sources. This is unique in the literature.

- Most trust algorithms just rate the act of cooperation. In contrast, this dissertation *makes the cooperation outcome the first class object*. The subjective likelihoods of the possible cooperation outcomes (named the trust distribution) should be predicted directly and as complete as possible. If necessary, a rating can be derived from them in a subsequent step, either in the trust algorithm or in a decision algorithm. The trust algorithm in ElSalamouny et al., 2010 is one of few examples that put out the cooperation outcome instead of a rating.

- Present trust algorithms compute specific trust for a certain purpose. The needs of a reputation system, for example, or the trust problem of an autonomous agent define that situation. The literature of the social sciences shows though that people can express trust for all sorts of attribute combinations like: the trust in a certain cooperation partner or the trust regarding a certain situational setting (e.g. meeting at night) (see Section 3.4.1). The Enfident Model resembles this with the *concept of querying*. This concept is unique in the technical literature. It enables a system to compute trust for a specific trust situation or to exchange the trust in various objects with other systems – with just one single trust model.

- The Enfident Model explicitly *models trust-related changes in the mentioned entities over time*. For example a cooperation partner could change its behaviour, which means its internal way of working, because of defects or software updates. I found a related functionality only recently in the literature: ElSalamouny et al. (2010) model the time-varying behaviour of a single cooperation partner as a hidden Markov model. The Enfident Model includes similar sub-models for

all entity types not just the cooperation partner and entangles those sub-models across entities. Moreover the Enfident Model proposes a time-dependent likelihood for the state transitions.

- Trust develops over an ordered sequence of acts of cooperation. An act of cooperation may in turn consist of an ordered sequence of interactions. The trustor can evaluate trust at any time during an act of cooperation. Some information may be known at that time, other information may be unknown and some information may change from interaction to interaction. To my knowledge, no present work contains such a *comprehensive sub-model for the temporal development during a single act of cooperation*.

- A trust mechanism should help to handle new, uncertain situational settings. Therefore it must *transfer knowledge* from other, even different settings to this new one by utilising similarities (Pan and Yang, 2010). Rettinger et al., 2008 is the only present work that realises this functionality satisfyingly.

The Enfident Model combines all these features in a coherent model and shows how they can interplay with each other. Present trust models focus on few of them only. This listing also clarifies why the Enfident Model can serve as a meta-model to analyse existing trust algorithms.

To realise this functionality, I propose *algorithms that combine clustering, learning, logic and probability theory in a relational dynamic Bayesian network* (e.g. Manfredotti, 2009). They are based on the algorithms in Xu, 2007 for static relational Bayesian networks and the algorithms in Van Gael, 2011 for infinite hidden Markov models.

For the evaluation, the Enfident Model is applied to the scenario of cooperating cognitive vehicles. This scenario features a whole “society” of self-organising systems. Since trust addresses a social problem, the evaluation with a realistic technical society matches best here. To my knowledge, such an evaluation is unique in the literature and was a complex undertaking.

## 1.4 Organisation

The organisation of this dissertation uses a methodology that follows the phases of a systematic engineering process with use cases, requirements, design, implementation and testing. At the same time, the text is organised in two parts: a generic and an application-specific part. To avoid duplication of text, some phases of the above process are detailed in one part or the other only, as described in the following.

*Problem definition and use cases.* Chapter 1 introduces the problem and sketches application scenarios. Chapter 2 then compiles views on trust from various fields to find a definition of trust and related terms for this dissertation. Those views and the definitions further clarify the problem. A comprehensive description of a single application together with use cases can be found in Chapter 8.

*Requirements.* Chapter 4 presents the requirements. They are based on a review of the socio-scientific literature on interpersonal trust in Chapter 3 and of the technical literature on trust in Chapter 4. Own considerations complement them.

*Design.* The requirements lead to an application- and implementation-independent design of a trust mechanism: the Enfident Model (Chapter 6). Chapters 4 and 6 together show that the Enfident Model suits as a framework to analyse existing technical trust algorithms and to design new ones. The preceding Chapter 5 introduces the notation of some mathematical tools that are used throughout the remainder of this document.

*Implementation.* Chapter 7 proposes implementation techniques for the Enfident Model. These techniques originate from statistical relational learning and are just implementation examples, because other techniques seem reasonable as well. Chapter 7 marks a first step towards a concrete algorithm. However the attributes are still unknown; they depend on the application. Chapter 8 then applies the model to a specific scenario. In this step, attributes can be identified and the algorithms can be completed.

*Test.* Chapter 8 describes the evaluation method. It introduces the application scenario of cognitive vehicles that cooperate through a vehicular network and defines the simulation environment. The evaluation results and the discussion are combined in Chapter 9, but separated in the subsections. In this way, one subject can be evaluated and discussed in one place, while the reader can still distinguish the results and their discussion.

Chapter 10 summarises the dissertation. For this purpose, it also relates the Enfident Model back to selected findings from social sciences. Finally it points out directions for future research.