TÉCNICO LISBOA

Trustful Action Suggestion in Human Agent Interaction

Nuno Miguel Xu Gonçalves

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Rui Prada
Prof. Ana Paiva

October 2016

Acknowledgments

I would like to thank my parents for their friendship, encouragement and caring over all these years, for always being there for me through thick and thin and without whom this project would not be possible. I would also like to thank my grandparents, aunts, uncles and cousins for their understanding and support throughout all these years.

Quisque facilisis erat a dui. Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus ornare, erat elit consectetuer erat, id egestas pede nibh eget odio. Proin tincidunt, velit vel porta elementum, magna diam molestie sapien, non aliquet massa pede eu diam. Aliquam iaculis.

Fusce et ipsum et nulla tristique facilisis. Donec eget sem sit amet ligula viverra gravida. Etiam vehicula urna vel turpis. Suspendisse sagittis ante a urna. Morbi a est quis orci consequat rutrum. Nullam egestas feugiat felis. Integer adipiscing semper ligula. Nunc molestie, nisl sit amet cursus convallis, sapien lectus pretium metus, vitae pretium enim wisi id lectus.

Donec vestibulum. Etiam vel nibh. Nulla facilisi. Mauris pharetra. Donec augue. Fusce ultrices, neque id dignissim ultrices, tellus mauris dictum elit, vel lacinia enim metus eu nunc.

I would also like to acknowledge my dissertation supervisors Prof. Some Name and Prof. Some Other Name for their insight, support and sharing of knowledge that has made this Thesis possible.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the good and bad times in my life. Thank you.

To each and every one of you – Thank you.

Abstract

Nulla facilisi. In vel sem. Morbi id urna in diam dignissim feugiat. Proin molestie tortor eu velit. Aliquam erat volutpat. Nullam ultrices, diam tempus vulputate egestas, eros pede varius leo, sed imperdiet lectus est ornare odio. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Proin consectetuer velit in dui. Phasellus wisi purus, interdum vitae, rutrum accumsan, viverra in, velit. Sed enim risus, congue non, tristique in, commodo eu, metus. Aenean tortor mi, imperdiet id, gravida eu, posuere eu, felis. Mauris sollicitudin, turpis in hendrerit sodales, lectus ipsum pellentesque ligula, sit amet scelerisque urna nibh ut arcu. Aliquam in lacus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla placerat aliquam wisi. Mauris viverra odio. Quisque fermentum pulvinar odio. Proin posuere est vitae ligula. Etiam euismod. Cras a eros.

Keywords

Maecenas tempus dictum libero; Donec non tortor in arcu mollis feugiat;Cras rutrum pulvinar tellus.

Resumo

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Aliquam aliquet, est a ullamcorper condimentum, tellus nulla fringilla elit, a iaculis nulla turpis sed wisi. Fusce volutpat. Etiam sodales ante id nunc. Proin ornare dignissim lacus. Nunc porttitor nunc a sem. Sed sollicitudin velit eu magna. Aliquam erat volutpat. Vivamus ornare est non wisi. Proin vel quam. Vivamus egestas. Nunc tempor diam vehicula mauris. Nullam sapien eros, facilisis vel, eleifend non, auctor dapibus, pede.

Palavras Chave

Colaborativo; Codificaçãoo; Conteúdo Multimédia; Comunicação;

Contents

List of Figures

List of Tables

List of Algorithms

Listings

Acronyms

P2P Peer-to-Peer

1

Introduction

Trust has been described in Psychology as being one of the most important components of interpersonal relationships [**?**]. It is undeniable the need of trust to promote cooperation and collaboration between two parties, either in deciding who should one collaborate with, or even on what exactly do we trust the other party with.

As **AI!** (**AI!**) research gravitates towards the development of Intelligent Agent Systems [**?**], where a focal concern is the performance of collaborative tasks [**?**, **?**, **?**], as well as addressing the problems of interaction between humans and agents [**?**], one would consider that trust should be one of the main focuses of **HAI!** (**HAI!**). Since the start of automated machinery, one of the main issues was how to properly manage trust on machines, in order to avoid over or under reliance [**?**]. Reeves and Nass have shown that people apply social rules to **HCI!** (**HCI!**), and this can logically be extended to the sub-field of **HAI!** [**?**]. So as agents evolve to better perform collaborative tasks with humans autonomously, which demands at least some amount of social interaction, the active agent must seek out to improve the trust relationship it has with the user [**?**]. And while the amount of literature has been increasing, we found it surprising that not enough work has been done in **HAI!** focusing on trust, other than on design issues [**?**] and the sub-field of **HRI!** (**HRI!**) [**?**, **?**], specially when so much has been done regarding **TiA!** (**TiA!**) [**?**, **?**, **?**]. This reveals that while the area has so much potential, the level of understanding is still very shallow, only deeply focused in certain areas [**?**].

MAS! (**MAS!**) trust and Reputation modelling is one of the areas that has been having a great increase of interest lately, specially ever since the advent of Peer-to-Peer (P2P) e-commerce in platforms like *eBay*[1]. For this applications, tools and solutions to ensure trust were needed for a new reality of a mass amount of anonymous entities constantly entering and exiting the environment and performing trading transactions through an open space. However almost all research focuses purely on the creation and maintenance of the internal trust model structure of the agent, normally with just the purpose of ranking other agents, through the use of statistical and game theoretical based methods [**?**]. This makes it difficult to create a model that is easy to understand, analyse and, most importantly, describe its evaluative reasoning in a human understandable manner. The introduction of cognitive models by Castelfranchi and Falcone [**?**] tries to solve that problem by mapping the trust model to the agent's mental state, composed by beliefs and goals, very akin to existing cognitive agent architectures like BDI [**?**]. Then some systems, like Repage [**?**], created implementations of this new paradigm of trust modelling; until then most of the models were purely theoretical. Nevertheless, there is a gap in this area of research that we wish to address with our work: the lack of an implementation for an action suggester based on the agent's trust model to improve the strength of our beliefs in the model and to improve trust in our agent. While one could argue that this is the responsibility of the decision making or planner component of the agent, we believe that a dedicated module will ease the complexity of decision by

[1] eBay Auctions: `http://www.ebay.com/`

making it more modular, and also allowing for a greater degree of integration with the trust model of the agent. To our knowledge, no attempts have been done towards this goal, so we propose to develop two agent modules: firstly, one capable of creating a cognitive model representing the mental state of the user's trust in the agent, using Repage's architecture, and secondly, another to suggest what actions should be used to improve trust on the agent. We will ascertain this project's objectives by integrating the modules in an agent implementation that is capable of acting as one of the players in the *Split or Steal* scenario, introduced in the British game show *Golden Balls*[2] (currently finishing development in our research group, **GAIPS!**[3]). The scenario is further described in Section **??**.

We hope that this project will make agent decision making more interesting, provide some insight on how actions affect trust and budge the field a bit in this unexplored direction.

In the remainder of the document we will present a brief summary of the main concepts used in this project in Section **??**. Then in Section **??**, we will discuss some of the work done in modelling trust for **MAS!**s and measuring trust in **HRI!** applications. Followed by a description of our solution architecture will be presented in Section **??**. Finally in Section **??** we will describe how we will evaluate the project.

[2] Golden Balls TV Show: http://www.goldenballstvshow.com/

[3] **GAIPS!** (**GAIPS!**): http://gaips.inesc-id.pt/

2

Background

Contents

Before discussing related work and our solution to the problem, we will present the main concepts that will be mentioned in the rest of this report, specifically regarding trust and reputation.

2.1 Trust

Trust is regarded throughout the literature as one of the fundamental components of human society, being essential in cooperative and collaborative behaviour, having been studied in a multitude of disciplines, from Psychology and Sociology, to Philosophy and Economy [**?**, **?**, **?**]. For that reason, it is no wonder that it acquired a very large number of different definitions throughout the years of study, causing the problem of not existing a consensus on a definition of trust [**?**]. In the scope of this project, the most relevant start for our discussion is the dyadic definition of trust: 'an orientation of an actor (the **truster**) toward a specific person (the **trustee**) with whom the actor is in some way interdependent' (taken from [**?**]), as we want to focus on interpersonal relationships. This definition has been expanded throughout the literature, often adapted to fit the context or scope of the work, but three main definitions are highlighted in computational trust:

- First, Gambetta [**?**] defined trust as follows: 'Trust is the *subjective probability* by which an individual, A, *expects* that another individual, B, performs a given action on which its *welfare depends*' (taken from [**?**]). This is accepted by most authors as one of the most classical definitions of trust, but it is too restrictive with its uni-dimensionality, as it only refers to predictability of the trustor, and does not take into account competence in executing the given action.

- Marsh [**?**] was the first author to formalize trust as a measurable Computational Concept, continuing the perspective of reducing trust to a numerical value, set by Gambetta [**?**], but also adding that: X trusts Y if, and only if, 'X *expects* that Y will behave according to X's best interest, and will not attempt to harm X' (taken from [**?**]). This definition does not represent other parts of trust, such as the notion that trustor must ascertain some risk from delegating the action to the trustee.

- Castelfranchi and Falcone then introduced a Cognitive aspect to Computational Trust [**?**]. They define trust as the mental state of the trustor and the action in which the trustor refers upon the trustee to perform. This is the definition of trust that we will adopt throughout the rest of the report, as it represents a vision of trust that takes into account the trustor set of beliefs and intentions, approaching it to an agent's cognitive model, while also linking trust to the action being performed, as one might trust another for certain types of actions and not for others (e.g. I may trust my squire to polish my sword, but not to swing it).

2.1.1 Castelfranchi and Falcone's Trust

More explicitly, Castelfranchi and Falcone [**?**] state that trust is a conjunction of three concepts:

- A *mental attitude* or (pre)disposition of the agent towards another agent; this is represented by beliefs about the trustees' qualities and defects;
- A *decision* to rely upon another, and therefore making the trustor 'vulnerable' to the possible negative actions of the trustee;
- The *act* of trusting another agent and the following behaviour of counting on the trustee to perform according to plan.

By describing trust as a mental attitude it is also implied that: 'Only a cognitive agent can trust another agent; only an agent endowed with goals and beliefs' [**?**].

From this definition we should also address one important component, **Delegation**, which happens when an agent (X) needs or likes the action delegated to another agent (Y), so X includes it in his plans, therefore relying on Y. X plans to achieve his goal through Y. So, he formulates in his mind a multi-agent plan with a state or action goal being Y's delegated [**?**].

2.2 Reputation and Image

Reputation is also a concept that appears very often linked with trust in the literature, specially since recent models created for representing trust have been focused on **MAS!**s (see [**?**, **?**, **?**, **?**, **?**]), where more recent trust models have been developed to also include reputation as a source of trust.

An agent is not influenced only by their own beliefs about the subject, the *Image*, but also by what other agents say about it, its *Reputation*.

We describe Image and Reputation as introduced by Sabater in [**?**]: Image is defined as the agent's personal belief about a certain property of the target agent, be it a physical, mental or social trait. Reputation is a meta-belief about an impersonal evaluation of the target, in other words, it is the belief on the evaluation being circulated about the target. On a more concrete level, reputation is separated between *shared evaluation* and *shared voice*. Consider that an agent has beliefs about how other agents evaluate a certain target, if in a set of agents these beliefs converge to a value (e.g. 'good' or 'bad') we can say that there exists a shared evaluation of the target. It is important to note that all sharing agents are known and well defined. A shared voice is a belief that another set of agents themselves believe that an evaluation of the target exists. In other words, it is the belief that a group of agents will consistently report that a voice exists. These meta-beliefs are considered important as one is not required to believe that other's evaluation is correct, but might still believe that it exists.

The mental decisions regarding reputation can be categorized as follows:

- Epistemic decisions: accepting trust beliefs to update or generate a given image or reputation;
- Pragmatic-Strategic decisions: using trust beliefs to decide how to behave towards other agents;
- Memetic decisions: transmitting trust beliefs to others.

This difference of possible decisions allows to describe how one may transmit reputation without having the responsibility for the credibility or truthfulness of the content transmitted, as one does not have to commit to accepting the reputation value, and just say that the rumour exists.

2.3 Game Theory

Game Theory is the field of study that defines and analyses situations involving conflict or cooperation between multiple intelligent decision makers. These situations are called a game, and they are distilled to their core argument, by defining the limited and simple set of actions that the players may perform, and how do they affect the players. It then analyses the decision strategies for each player, by assuming that both will try to maximise their payoff (how much the player gains) with their action. To better explain the concepts we want to present, we will introduce one of the most common exemplary models of Game Theory, the Prisoner's Dilemma.

2.3.1 Prisoner's Dilemma

The Prisoner's Dilemma is a two player game and is usually described as follows:

Two criminal partners are arrested and locked in separate cells with no way of communicating with each other. They are then questioned separately, where they are given 2 options, betray the other prisoner by testifying against him, or remain silent, with the following outcomes:

- If both prisoners betray each other, both get 2 years in prison;
- If one of them betrays and the other remains silent, the betrayer goes free and the other gets 3 years in prison;
- If both remain silent, both get just 1 year in prison;

We can represent betraying as *Defecting* (D), and staying silent as *Cooperating* (C), and name the players *player1* and *player2*. So the game's possible outcomes can be represented by a payoff matrix, like the one in Table 2.1, where each entry represents a tuple of the form (*player1* payoff, *player2* payoff). As the goal is to not get years in prison, the payoffs correspond to $Max\ years\ in\ prison - years\ got\ in\ prison$.

In the game we can say that *Defecting* **dominates** *Cooperating*, as for any action that the adversary player may choose, *Defecting* always gives a better payoff for the individual player [**?**].

	C_2	D_2
C_1	2,2	0,3
D_1	3,0	1,1

Table 2.1: Prisoner's Dilemma Payoff Matrix

3

Related Work

Contents

Computational Trust research has been focused on modelling trust in **MAS!**s, specially on open e-commerce environments [**?**, **?**, **?**, **?**, **?**], with at least 106 models created [**?**], since the formalization of trust as a measurable property by Marsh in 1994 [**?**]. We will present some trust models from which we will take inspiration while creating our own, and some work done in measuring trust in **HRI!**.

3.1 Trust Models

For related work concerning Trust Models we will focus on **Cognitive** Trust Models, first introduced by Castelfranchi and Falcone [**?**], which are defined by measuring trust on the strength of an agent's beliefs and the changes enacted through the consequent act of trusting. We want to focus on modelling trust through multiple dimensions, with the intent of having trust depend on the action to perform, context and agent performing the task and having these dimensions represented explicitly in the model, something that it is not possible with **Numerical** models, like the one introduced by [**?**].

3.1.1 Castelfranchi and Falcone's model

Having developed the concept of Cognitive Trust Models, this author's model is generally regarded as a classical basis for most other authors, and while we will not use the entirety of this model, it is worth describing, as it was also a source of inspiration to other authors referenced in this report. The model is characterised around their definition referred in Section **??**, through a central core, composed by a five-part relation, between:

- The trustor (**X**);
- The trustee (**Y**);
- The context where they are inserted in (**C**);
- A task ($\boldsymbol{\tau}$) defined by the pair (α, ρ), where $\boldsymbol{\alpha}$ is the action entrusted to the trustee, that possibly produces an outcome $\boldsymbol{\rho}$, contained in the goal of X (g_x);
- The goal of the trustor ($\boldsymbol{g_x}$).

More shortly represented by equation 3.1.

$$TRUST(X\ Y\ C\ \tau\ g_x) \tag{3.1}$$

This defines Trust as goal-oriented, contextual, and multi-dimensional, as from the point of view of the trustor, it varies not only on the trustee, but also from the overall context, the action that is being delegated, and the particular goal of the trustor. For example, if the goal of the trustor is simple to perform and not very critical to him, he may be more willing to delegate the task, and trust another agent to perform such task. Adjustments can be attached to this core adjusting better to the context in which it may be used. For instance, one may add an authoritative third party element to the relation in supervised security applications.

The model also conceptualizes **Expectation** as a belief of when agent X awaits for ρ to happen when an action α trusted to Y is being performed, formalized in first order logic in equation 3.2.

$$(\textit{Expectation}\ X\ \rho) \implies (Bel_x^{t'}(\textit{will-be-true}^{t''}\rho)) \wedge (Goal_x^{Period(t',t''')} (KnowWhether_X(\rho\ OR\ \textit{Not}\ \rho)^{t''})) \tag{3.2}$$

This can be used to establish what expectations the user should have in the agent, whether initial or constructed during interaction, and provide an additional measure to weight the importance of certain agent functions and actions.

As stated in the definition (Section **??**) the mental attitude of the trustor X is defined by beliefs of the qualities (and faults) of Y. Therefore we can quantify the strength of our belief in a certain quality through its **DoC! (DoC!)**, which is defined by a function **F** that takes all different belief sources for this quality, as shown in equation 3.3, where for a source sj, Str_j represents the value of the source and *Qual-i*$_{sjY}(\tau)$ the value of quality i of agent Y provided by the source in performing task τ.

$$DoC_X(\textit{Qual-i}_{(s1,...sn),Y}(\tau)) = F_{X,Y,\tau}(Bel_X(Str_1\textit{Qual-i}_{s1Y}(\tau)), Bel_X(Str_2\textit{Qual-i}_{s2Y}(\tau)), ..., Bel_X(Str_n\textit{Qual-i}_{snY}(\tau))) \tag{3.3}$$

$F_{X,Y,\tau}$ associates the *strengh-of-sources (Str_j)* and *quality-values (Qual-i$_{sjY}(\tau)$)* with a probability curve. It should return a matrix with two columns, with an amount of rows corresponding to the number of quality values selected out of the received as input (since not all values must or should be used, and some may be integrated into a single value), and the first column should contain these values associated with their normalized probabilities in the second column (the probabilities sum should be 1).

For example, consider that we want agent X's **DoC!** regarding Y's ability to clean:

- We have two sources about Y's ability to clean:
  1. X saw Y once clean quite well, but long ago, so we could attribute *Ability*$_{s1Y}(cleaning) = 0.8$ and $Str_1 = 0.2$;
  2. Someone X considers reliable informs that Y performed poorly recently, se we attribute *Ability*$_{s2Y}(cleaning) = 0.2s$ and $Str_2 = 0.6$;
- So a possible result of $DoC_X(\textit{Ability}_Y(cleaning))$ is:

$$\begin{pmatrix} 0.8 & 0.25 \\ 0.2 & 0.75 \end{pmatrix}$$

Finally **DoT! (DoT!)** quantifies the Trust level agent X has in Y to perform task τ according to the formula depicted in equation 3.4.

$$\begin{aligned} DoT_{XY\tau} = c_{Opp}\ DoC_x[Opp_y(\alpha,\rho)] \times \\ \times\ c_{Ability_y}\ DoC_x[Ability_y(\alpha)] \times \\ \times\ c_{WillDo}\ DoC_x[WillDo_y(\alpha,\rho)] \end{aligned} \tag{3.4}$$

Where:

- $DoC_x[Opp_y(\alpha,\rho)]$ is the **DoC!** of X's beliefs about all contextual factors in which Y will act; in other words, the degree of Opportunity Y has to do α and result in ρ;
- $DoC_x[Ability_y(\alpha)]$ is the **DoC!** of X's beliefs about Y's ability to perform α;
- $DoC_x[WillDo_y(\alpha,\rho)]$ is the **DoC!** of X's beliefs concerning if Y's actually is going to perform α with the result ρ;
- c_{Opp}, $c_{Ability_y}$ and c_{WillDo} are constants representing the weight of each **DoC!**.

This model is the most abstract, as almost all of the implementation details are left aside, particularly how the beliefs are modelled and how to or even what should be a good quantification to the quality values for the agent. This provides a lot of liberty on how to contextualize the model, and for our modules such adaptability is interesting for our intent to try our modules in different scenarios.

3.1.2 Repage: A REPutation and ImAGE model

This system was introduced in 2006 by Sabater et al. [**?**] and aims to establish two different aspects to trust modelling, Image and Reputation, as defined in Section **??**. The representation for an evaluation are fuzzy sets, defined by a tuple of five positive numbers(summing to one), where each number corresponds to a value of probability (weights) traced directly to the following scale: *very bad (VB), bad (B), neutral (N), good (G), very good (VG)*. Additionally the strength of the belief is added to the tuple, so it can be represented like this $\{w_1, w_2, ..., w_5, s\}$.

The architecture is composed by three main elements, a *memory*, a set of *detectors*, and the *analyser* (check Figure 3.1). Memory is composed by predicates that are conceptually organized in different levels of abstraction and are inter-connected by a network of dependencies that propagate changes and inferences through the various predicates. The predicates contain a fuzzy evaluation belonging to one of the following types (image, reputation, shared voice, shared evaluation, valued info, evaluation from informers, and outcomes), and refer to a certain agent performing a specific role. The detectors infer new predicates, remove non-useful ones and builds the dependency network.

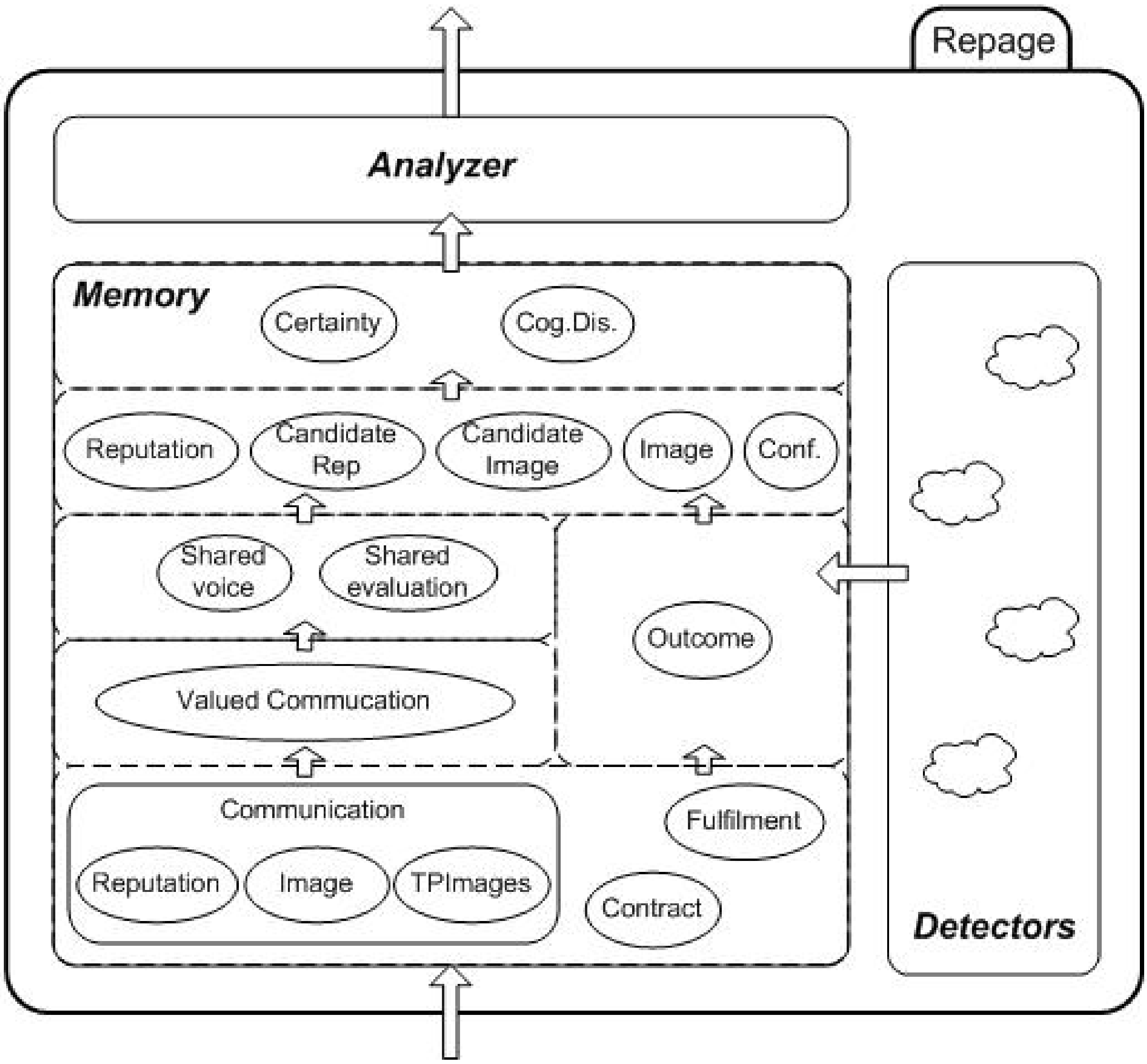


Figure 3.1: Repage architecture schematic (taken from [**?**])

At the first level of the abstraction hierarchy we have the basis of information to infer predicates, *contracts*, *fulfilments* and *communication* (they are not themselves predicates, as no evaluation is attached). Contracts are agreements between two agents, while fulfilments are the results of the contract. Communication is the information about other agents that come from third parties. The second level is then constituted by inferences to an outcome, formed by a contract and its fulfilment, and valued information gathered from communications. This inferred predicates are not just tuples, they give an evaluation to the predicate, setting its belief strength.

In the next level we have two predicates: *shared voice* and *shared evaluation*. The former is inferred from communicated reputation, and the latter from communicated images.

The fourth level is composed from five types of predicates: *Candidate Image*, *Candidate Reputation*, *Image*, *Reputation* and *Confirmation*. The candidate predicates are Images and Reputations that do not have enough support yet. Special detectors turns them to fill image/reputations when a strength threshold is surpassed. Confirmation is the feedback to a communication, received from comparing it to the image of the target.

Finally the last abstractions level is composed of the predicates *cognitive dissonance* and *certainty*. Cognitive dissonance is a contradiction between relevant pieces of information that refer to the same target. This predicate may create instabilities in the mind of the individual, so the agent will most likely try to perform action in order to confirm the sources of this dissonance. Certainty represents full reliance on what the predicate asserts.

The last element is the analyser and its job is to propose actions in order to improve the accuracy of predicates in Repage and solve cognitive dissonances to produce certainty. The actions are proposed to the agent planner, leaving it to decide how to take this actions into account.

Image and Reputation are the predicates that provide a trust evaluation of a target, and as previously stated, they have a role, that represents two things: the agents interaction model, in other words, the actions that may affect to this evaluation, and a function that contextualizes the evaluative labels of *VB, B, N, G, VG*. The probability distribution of the values gives out a picture of the target interaction forecast (e.g. a probability value of 0.5 to VB gives a 50% chance of the next interaction with the target being very bad).

The work described here is the only found that tries to establish an implementable architecture for a trust model, as most of the models created are purely theoretical. Furthermore, it fits to our goals of creating a trust assessment module, corresponding to the memory and detector components, and a trust decision module, corresponding to the analyser.

3.1.3 *BC*-logic: A Representation of Beliefs for Repage

Pinyol et al. [**?**] proposes an integration of the Repage model, introduced in the previous Section **??** with a BDI Agent [**?**]. While the BDI model is not relevant to us, their work specifies *BC*-logic, a belief first order logic that is capable of representing Repage predicate semantics and this is the part we will describe in the following paragraphs.

BC-logic is structured hierarchically, in a way that formulas from a certain first-order language lower in the hierarchy can be embedded in another language above as constants. This is written as $\lceil \phi \rceil$, with ϕ being the formula of the lower language. The hierarchy is composed of three languages, starting with the base language, L_{basic}, that expresses the ontology and contains the symbols to represent the domain. Next there is L_{ag}, which contains symbols of the base language and special predicates to allow to reason about probability of formulas, about formulas communicated, and formulas believed by agents. Finally there is *BC*-language, the meta-logic language, with the aim to express statements about the agents' reasoning. L_{ag} and *BC*-language are sorted languages, in other words, its symbols and predicates are partitioned into sorts, each containing their own semantics. All languages contain the logical symbols $\forall$, $\exists$, $\wedge$, $\vee$, $\neg$, and $\implies$.

L_{ag} contains four sorts:

- S_D: represents application domain, including constants, functions and predicate symbols;
- S_R: represents probabilities, including a set of constants, C_R, with a label written as $\overline{r}$, where $r \in [0, 1] \bigcap Q$; Q being the set of rational numbers;
- S_A: represents agent names, including a set of constants $C_A = i_1, ..., i_n$, corresponding to the agents' identifiers;
- S_F: represents formulas, including a set of constants C_F, which is built simultaneously with the construction of the language. This is done by adding the constant $\lceil \phi \rceil$ for each $\phi \in F_m(L_{basic})$, and then, given a formula $\Psi \in F_m(L_{ag})$ we also add $\lceil \Psi \rceil$ to C_f.

Symbols in predicates are identified by their sorts, take for example a binary predicate B, it is written as $B(S_A, S_F)$, meaning that the first argument must be part of sort S_A and the second argument part of sort S_F.

The set of formulas $Fm(L_{ag})$ has the following special predicates:

- $B(S_A, S_F)$: An agent's belief towards a formula (e.g. $B(i_c, \lceil sunny(Lisbon) \rceil)$); abbreviated to $B_{S_A}(S_F)$;
- $Pr \leq (S_F, S_R),\ Pr \geq (C_F, C_R)$: A lower/upper bound on probability of a formula (e.g. $Pr \geq (\lceil sunny(Lisbon) \rceil, 0.8)$);
- $S(S_A, S_A, S_F)$: The communication predicate, as stated in Repage (e.g. $S(i_c, j_c, \lceil sunny(Lisbon) \rceil)$); abbreviated to $S_{S_A, S_A}(S_F)$.

BC-language contains five sorts:

- S_R, S_A and S_F: as defined above for L_{ag};
- S_V: represents variable sequences;
- S_T: represents ground term sequences;

Image and Reputation towards an agent j_c, playing the role r can then be represented by:

- Image: $Img_{i_c}(j_c, r, [V_{w_1}, ..., V_{w_m}])$
- Reputation: $Rep_{i_c}(j_c, r, [V_{w_1}, ..., V_{w_m}])$

$$\frac{Img_{i_c}(j_c, r, [V_{w_1}, ..., V_{w_m}])}{\begin{array}{c} B_{i_c}(pr \geq ([R_{rj_c} T_{r_{w_1}}, V_{w_1}])) \\ B_{i_c}(pr \geq ([R_{rj_c} T_{r_{w_2}}, V_{w_2}])) \\ ... \end{array}} \qquad \frac{Rep_{i_c}(j_c, r, [V_{w_1}, ..., V_{w_m}])}{\begin{array}{c} B_{i_c}(s(pr \geq ([R_{rj_c} T_{r_{w_1}}, V_{w_1}]))) \\ B_{i_c}(s(pr \geq ([R_{rj_c} T_{r_{w_2}}, V_{w_2}]))) \\ ... \end{array}}$$

with $[V_{w_1}, ..., V_{w_m}]$ being an abstracted set of evaluations in the belief, as while Repage maps evaluation from Very Bad to Very Good, this can be applied to any ordered mapping of m evaluations. As a simplification, the model summarizes the interaction model of the participating agents (i_c and j_c) to a single action. Through this a mapping $R_r i$ can be defined between each role r, agent i_c and the action. A mapping T_{r,w_k} is also defined between each role r and label w_k to a formula written in L_{basic}.

Image and Reputation can also be represented as a set of beliefs:

The work goes into further detail regarding representing the relationship of Image and Reputation, addressing agent honesty and consistency in communicating reputation, and while interesting, it is not part of what we to model.

Overall *BC*-logic is an interesting approach to representing beliefs in the Repage model and we will most likely choose it for the model representation, as it is the most well developed that we found.

3.1.4 Sutcliffe and Wang's model

This work was published by Sucliffe and Wang in 2012 [**?**] and they built a trust model to figure out how cognitive social mechanisms emerge to follow Dunbar's **SBH!** (**SBH!**) [**?**], an evolutionary psychology theory that proposes that humans have a predisposition to build relationships in layers of decreasing intimacy. As trust has been acknowledged to be one of major component of human relationships, they demonstrate that simulating trust development and decay, through interactions and neglect, respectively, show the patterns predicted in **SBH!**.

From the model standpoint, its main interesting feature is that agents develop trusting relationships between one another, affecting interaction frequency between agents, by preferring to pick those with already high trust value. Additionally the trust relationship degrades as time passes by, with variable speeds depending on the current relationship level, giving stronger ties a slower descent. All in all, the model provides a good tool to simulate multi agent social behaviour, and may be interesting to predict trust degradation in the agent, albeit its described application for social simulations is a bit far from our scope.

3.1.5 Discussion

Of the related work discussed here, we are going to base our solution on Repage and *BC*-logic, as described in their respective Sections **??** and **??**. Repage fits well as a basis for our objectives, as it has the details of modelling trust already dealt with and leaves us the room to develop the analysis component that corresponds directly to the goal of this project. The choice was also made out of convenience, as no other work was found were implementable design was a concern.

3.2 The Perception and Measurement of Human-Robot Trust

Schaefer [**?**] presents a trust perception scale providing a way of extracting an accurate trust score from humans interacting with robots. The scale is composed of 40 items that can be ranked from 0 to 100, in 10 point intervals. The final result it then averaged by adding all the item values and divided by the total number of items (40).

While this work has been done specifically for **HRI!** we believe that a sub-set of this items can be used for the features used in the cognitive model of the user's trust, further described in Section **??**. The items are listed in Table **??** in appendix **??**.

4

Trust Model

Contents

We sought out to develop a trust model definition that would be easily implementable, but generic enough to be able to adapt to various testing scenarios. To do this we took inspiration from the work by Sabater et al. [**?**] described in Section **??** by taking a similar approach to architecture where a central memory component holds the model's current state, getting updated by perceptions received from the environment. But while Repage describes a third module that suggests actions to resolve belief conflicts in the model, we instead defined such module to assume the point of view of one of the agents in the scenario and, if participating in a social interaction, it suggests actions to improve the trust relationship with a trustor. In fact, most of the design of the model was made with the intent that it would be used by one of the agent's in the scenario, and the model created would be his own trust model of the world environment. And so, the model can be described by 3 main components:

- **Memory**, which defines and stores the main model structure;
- **Perceptions**, a series of environment perceptions mapped to changes in the Memory;
- **Action Suggestion**, a module that outputs different actions depending on current perceptions and the state of the model.

4.0.1 Memory

One of the main concerns while designing the model was how trust would be calculated, as we wanted to use Castlefranchi and Falcone's conceptualization of trust [**?**] as a basis for trust definition, focusing specially on it being dependent on the task entrusted, and the transferability of trust between different tasks. But starting from the five-part definition of trust, as seen in Equation 3.1, we decided that inserting context (**C**) and the trustor's goal (g_x) into the model would bring in too much complexity for the scope of this thesis, as it would require for a world state model to be kept, as well as some way to predict the trustor's goal. So we simplified, defining trust through a simpler three-part relation, involving just the trustor (**X**), the trustee (**Y**) and the task (τ), represented in Equation 4.1.

$$TRUST(X\ Y\ \tau) \tag{4.1}$$

So we designed the structure with the concepts and relations represented in Figure 4.1, and we can describe them as follows:

- **Agent**: a simple representation of the known entities in the scenario world space, serving mostly as an identifier;
- **Trustee**: each agent contains a collection of other agents he has information about, either by reputation, or by interaction, which we represent as their Trustees;

- **Trust Feature**: a piece of information a trustor has on a trustee is represented in a Trust Feature, which contains the Belief Sources of said information. The Feature Model defines and uniquely identifies what feature is represented.

- **Feature Model**: the possible set of trust features from which a trustee can be assigned is defined in a collection of Feature Models where each one represents a possible piece of trust related information relevant to the model scenario (e.g. The trustee's ability to cook, or the willingness to drive);

- **Category**: a Feature Model must belong to a Category, making it easier to present the different type of Trust Features;

- **Belief Source**: this represents a source of information on the corresponding feature, belonging to one of the 3 sub-classes depending on the origin of the information, Reputation for when reported from other agents (whether directly (e.g. talking) or indirectly (e.g. report on newspaper)), Bias for pre-existing beliefs on the feature, and Direct Contact for direct observations of the trustee, 3 values are provided to determine the associated feature's belief value:
  - Belief Value, a number between 0.0 and 1.0 describing the trustor evaluation;
  - Certainty describes how well the trustee was evaluated, in Reputation for instance, this might represent how well we trust in the reporter, and in Direct Contact how well the trustor observed the trustee performing said feature;
  - Time is just a record of when was this belief source recorded, as older records might have a lower impact in the overall belief value score, compared to newer records.

- **Task**: a representation of the possible delegation tasks in the scenario, containing the Feature Models associated with the performance of this task (e.g. The ability to serve drinks if the task is bartending). A weight is given to each Feature corresponding to its importance in the task. The various weights are normalized so that their sum is 1.0.

4.0.1.A Trust Calculation

Taking a Trustor X, a Trustee Y and a delegated task τ, Trust can then be calculated by taking the Trustee's Trust Features F_y, the Task's Feature Models F_τ and checking which they have in common, which we can represent as $F_{y \cap \tau}$. Remember that Trust Features are uniquely identified by a Feature Model. So after getting $F_{y \cap \tau}$ we can apply a linear function to each of the features in $F_{y \cap \tau}$, where for each element F_i we multiply the trustee's feature's belief value $B(F_i)$ with the weight of the feature for

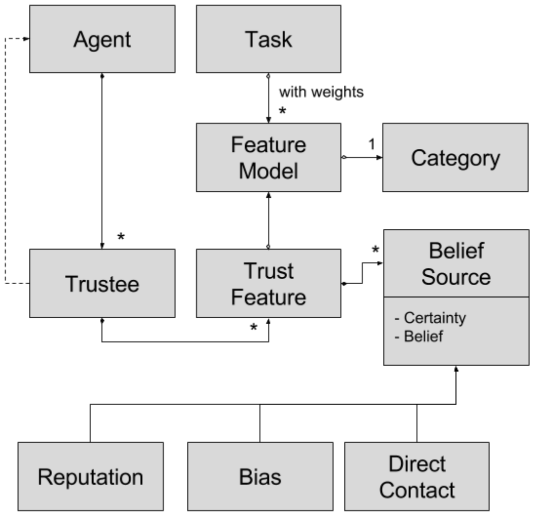


Figure 4.1: Memory Architecture (represented in UML)

the task $W(F_i)$, as represented in Equation 4.2.

$$Trust_{X,Y,\tau} = \sum_{i=0}^{n} W(F_i)B(F_i) \tag{4.2}$$

The belief value of the feature itself, $B(F_i)$, is also calculated through a sum of parameters pertaining to each of the n belief sources $B^j_{F_i}$ composing the feature, as represented in Equation 4.3, with each parameter described as follows:

$$B(F_i) = \sum_{j=0}^{n} D^j_{F_i} C^j_{F_i} B_j \tag{4.3}$$

- $D^j_{F_i}$, a value from 0.0 to 1.0 that represents how far ago in time was this belief source received compared to the last one, being 0.0 a long time ago, and 1.0 the most recent belief. We wished to represent the rapid decay of value of old beliefs when compared to new ones, but also making sure recent memories would not fall quickly in value, so we chose to describe this parameter with a Gaussian Function, as represented in Equation 4.4, where $T^{Last}_{F_i}$ is the most recent belief value's time stamp, $T^j_{F_i}$ is $B^j_{F_i}$ belief value's time stamp, and L is the difference between the oldest and newest belief value's time stamps. $\frac{L}{4}$ defines the mid drop-off point of the function.

- $C^j_{F_i}$, the certainty value stored in the Belief Source;

- $B^j_{F_i}$, the belief value stored in the Belief Source;

$$D^j_{F_i} = e^{-\frac{T^{Last}_{F_i} - T^j_{F_i}}{2(\frac{L}{4})^2}} \tag{4.4}$$

4.0.2 Perception

In this module, a collection of relevant environment perceptions is inserted into the model, in order to translate perceived changes in the environment, into changes in the model. This is done through a Perception object, representing some possible environment input, and containing a map of what target features should have belief sources added, what kind of belief sources they are, and how to translate the values received from the environment to belief value and certainty, as exemplified in Figure 4.2.

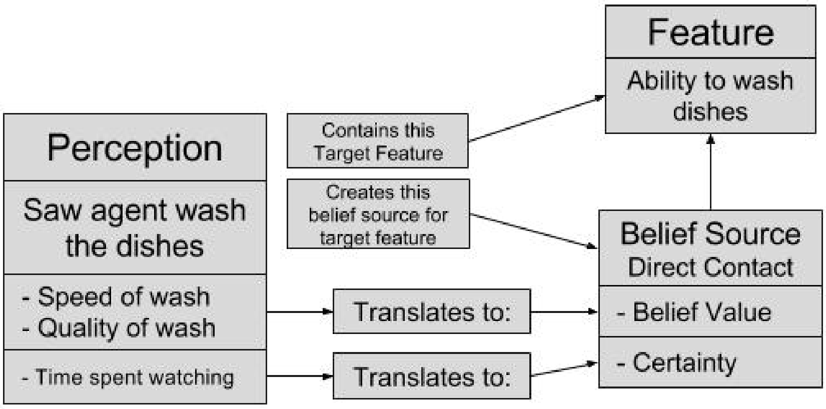


Figure 4.2: Perception Example

4.0.3 Action Suggestion

This component contains a collection of

5

This is the Fifth Chapter

Contents

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

5.1 Maecenas vitae nulla consequat

Aliquam aliquet, est a ullamcorper condimentum, tellus nulla fringilla elit, a iaculis nulla turpis sed wisi. Fusce volutpat. Etiam sodales ante id nunc. Proin ornare dignissim lacus. Nunc porttitor nunc a sem. Sed sollicitudin velit eu magna. Aliquam erat volutpat. Vivamus ornare est non wisi. Proin vel quam. Vivamus egestas. Nunc tempor diam vehicula mauris. Nullam sapien eros Fig. 5.1, facilisis vel, eleifend non, auctor dapibus, pede.

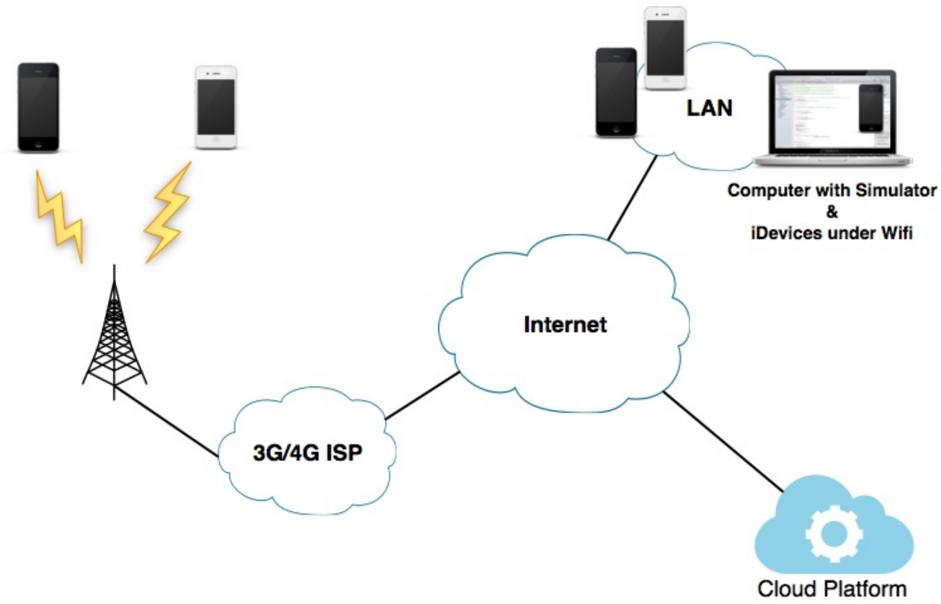


Figure 5.1: Test Environment

Aliquam aliquet, est a ullamcorper condimentum, tellus nulla fringilla elit, a iaculis nulla turpis sed wisi. Fusce volutpat. Etiam sodales ante id nunc. Proin ornare dignissim lacus. Nunc porttitor nunc a sem. Sed sollicitudin velit eu magna. Aliquam erat volutpat. Vivamus egestas. Nunc tempor diam vehicula mauris. Nullam sapien eros, facilisis vel, eleifend non, auctor dapibus, pede Table 5.1 used in the tests. The Network Link Conditioner allows to force/simulate fluctuations in fixed network segments.

Table 5.1: Network Link Conditioner Profiles

Network Profile	Bandwidth	Packets Droped	Delay
Wifi	40 mbps	0%	1 ms
3G	780 kbps	0%	100 ms
Edge	240 kbps	0%	400 ms

Aliquam aliquet, est a ullamcorper condimentum, tellus nulla fringilla elit, a iaculis nulla turpis sed wisi. Fusce volutpat. Etiam sodales ante id nunc. Proin ornare dignissim lacus. Nunc porttitor nunc a sem. Sed sollicitudin velit eu magna. Aliquam erat volutpat. Vivamus ornare est non wisi. Proin vel quam. Vivamus egestas. Nunc tempor diam vehicula mauris. Nullam sapien eros, facilisis vel, eleifend non, auctor dapibus, pede.

5.2 Proin ornare dignissim lacus

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis.

Et "optimistic" nulla dui purus, eleifend vel, consequat non, dictum porta, nulla. Duis ante mi, laoreet ut, commodo eleifend, cursus nec, lorem. Aenean eu est. Etiam imperdiet turpis. Praesent nec augue. Curabitur ligula quam, rutrum id, tempor sed, consequat ac, dui G_j, nec ligula et lorem consequat ullamcorper p ut mauris eu mi mollis luctus j, porttitor ut, formula 5.1, uctus posuere justo:

N_j Is the number of times peer j has been optimistically unchoked.

n_j Among the N_j unchokes, the number of times that peer j responded with unchoke or supplied segments to peer p.

$C_{r[j]}$ The cooperation ratio of peer j. If peer j never supplied peer p, the information of $C_{r[j]}$ may not be available.

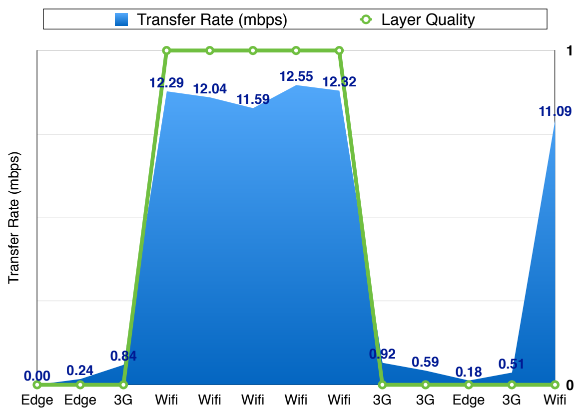
$C_{r(max)}$ The maximum cooperation ratio of peer p's neighbors, i.e., $C_{r(max)} = max(C_r)$.

$$G_j = \begin{cases} \dfrac{n_j C_{r[j]}}{N_j} & \text{if } n_j > 0 \\ \dfrac{C_{r(max)}}{N_j + 1} & \text{if } n_j = 0 \end{cases} \tag{5.1}$$

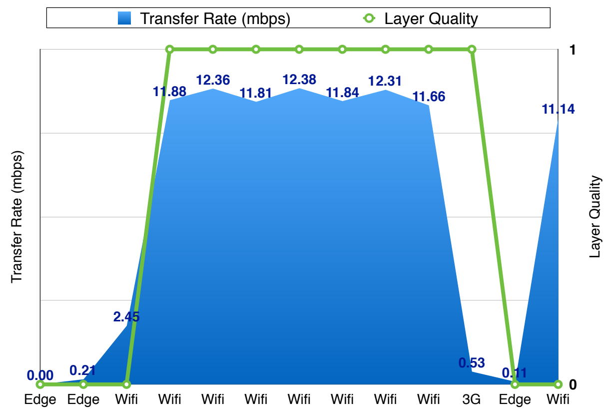
Cursus $C_{r(max)}$ conubia nostra, per inceptos hymenaeos j gadipiscing mollis massa $N_j = 0$, unc ut dui eget nulla venenatis aliquet $G_j = C_{r(max)}$.

Vestibulum accumsan eros nec magna. Vestibulum vitae dui. Vestibulum nec ligula et lorem consequat ullamcorper. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Phasellus eget nisl ut elit porta ullamcorper. Maecenas tincidunt velit quis orci. Sed in dui. Nullam ut mauris eu mi mollis luctus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Sed cursus cursus velit. Sed a massa.

Both Fig. 5.2(a) et Fig. 5.2(b) Phasellus eget nisl ut elit porta “perfect” tincidunt. Class aptent taciti sociosqu ad litora torquent per conubia nostra.



(a) Adaptation System Test 4



(b) Adaptation System Test 5

Figure 5.2: Adaptation System Behavior Test

Cras sed ante. Phasellus in massa. Curabitur dolor eros, gravida et, hendrerit ac, cursus non, massa. Aliquam lorem. In hac habitasse platea dictumst. Cras eu mauris. Quisque lacus. Donec ipsum. Nullam vitae sem at nunc pharetra ultricies. Vivamus elit eros, ullamcorper a, adipiscing sit amet, porttitor ut, nibh. Maecenas adipiscing mollis massa. Nunc ut dui eget nulla venenatis aliquet. Sed luctus posuere justo. Cras vehicula varius turpis. Vivamus eros metus, tristique sit amet, molestie dignissim, malesuada et, urna.

6

Conclusion

Contents

Pellentesque vel dui sed orci faucibus iaculis. Suspendisse dictum magna id purus tincidunt rutrum. Nulla congue. Vivamus sit amet lorem posuere dui vulputate ornare. Phasellus mattis sollicitudin ligula. Duis dignissim felis et urna. Integer adipiscing congue metus.

6.1 Conclusions

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Quisque facilisis erat a dui. Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus ornare, erat elit consectetuer erat, id egestas pede nibh eget odio. Proin tincidunt, velit vel porta elementum, magna diam molestie sapien, non aliquet massa pede eu diam. Aliquam iaculis. Fusce et ipsum et nulla tristique facilisis. Donec eget sem sit amet ligula viverra gravida. Etiam vehicula urna vel turpis. Suspendisse sagittis ante a urna. Morbi a est quis orci consequat rutrum. Nullam egestas feugiat felis. Integer adipiscing semper ligula. Nunc molestie, nisl sit amet cursus convallis, sapien lectus pretium metus, vitae pretium enim wisi id lectus. Donec vestibulum. Etiam vel nibh. Nulla facilisi. Mauris pharetra. Donec augue. Fusce ultrices, neque id dignissim ultrices, tellus mauris dictum elit, vel lacinia enim metus eu nunc.

Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam elementum condimentum feugiat, tellus odio consectetuer wisi, vel nonummy sem neque in elit. Curabitur eleifend wisi iaculis ipsum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam. Curabitur sed leo.

Nulla facilisi. In vel sem. Morbi id urna in diam dignissim feugiat. Proin molestie tortor eu velit. Aliquam erat volutpat. Nullam ultrices, diam tempus vulputate egestas, eros pede varius leo, sed imperdiet lectus est ornare odio. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Proin consectetuer velit in dui. Phasellus wisi purus, interdum vitae, rutrum accumsan, viverra in, velit. Sed enim risus, congue non, tristique in, commodo eu, metus. Aenean tortor mi, imperdiet id, gravida eu, posuere eu, felis. Mauris sollicitudin, turpis in hendrerit sodales, lectus ipsum pellentesque ligula, sit amet scelerisque urna nibh ut arcu. Aliquam in lacus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla placerat aliquam wisi. Mauris viverra odio. Quisque fermentum pulvinar odio. Proin posuere est vitae ligula. Etiam euismod. Cras a eros.

Nunc auctor bibendum eros. Maecenas porta accumsan mauris. Etiam enim enim, elementum sed, bibendum quis, rhoncus non, metus. Fusce neque dolor, adipiscing sed, consectetuer et, lacinia sit amet, quam.

6.2 System Limitations and Future Work

Aliquam aliquet, est a ullamcorper condimentum, tellus nulla fringilla elit, a iaculis nulla turpis sed wisi. Fusce volutpat. Etiam sodales ante id nunc. Proin ornare dignissim lacus. Nunc porttitor nunc a sem. Sed sollicitudin velit eu magna. Aliquam erat volutpat. Vivamus ornare est non wisi. Proin vel quam. Vivamus egestas. Nunc tempor diam vehicula mauris. Nullam sapien eros, facilisis vel, eleifend non, auctor dapibus, pede.

Bibliography

A

Code of Project

Nulla dui purus, eleifend vel, consequat non, dictum porta, nulla. Duis ante mi, laoreet ut, commodo eleifend, cursus nec, lorem. Aenean eu est. Etiam imperdiet turpis. Praesent nec augue. Curabitur ligula quam, rutrum id, tempor sed, consequat ac, dui. Vestibulum accumsan eros nec magna. Vestibulum vitae dui. Vestibulum nec ligula et lorem consequat ullamcorper.

Listing A.1: Example of a XML file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<StreamInfo version="2.0">
    <Clip duration="PT01M0.00S">
        <BaseURL>videos/</BaseURL>
        <Description>svc_1</Description>
        <Representation mimeType="video/SVC" codecs="svc" frameRate="30.00" bandwidth="401.90"
            width="176" height="144" id="L0">
            <BaseURL>svc_1/</BaseURL>
            <SegmentInfo from="0" to="11" duration="PT5.00S">
```

```
                <BaseURL>svc_1-L0-</BaseURL>
            </SegmentInfo>
        </Representation>
        <Representation mimeType="video/SVC" codecs="svc" frameRate="30.00" bandwidth="1322.60"
            width="352" height="288" id="L1">
            <BaseURL>svc_1/</BaseURL>
            <SegmentInfo from="0" to="11" duration="PT5.00S">
                <BaseURL>svc_1-L1-</BaseURL>
            </SegmentInfo>
        </Representation>
    </Clip>
</StreamInfo>
```

Etiam imperdiet turpis. Praesent nec augue. Curabitur ligula quam, rutrum id, tempor sed, consequat ac, dui. Maecenas tincidunt velit quis orci. Sed in dui. Nullam ut mauris eu mi mollis luctus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Sed cursus cursus velit. Sed a massa. Duis dignissim euismod quam.

Listing A.2: Assembler Main Code.

```
; **********************************************************************
; * Constantes
; **********************************************************************

ON     EQU  1 ; contagem ligada
OFF    EQU  0 ; contagem desligada
INPUT  EQU  8000H ; endereço do porto de entrada
          ;(bit 0 = RTC; bit 1 = botão)
OUTPUT   EQU  8000H ; endereço do porto de saída.


; **********************************************************************
; * Stack
; **********************************************************************

PLACE    1000H
pilha:     TABLE  100H   ; espaço reservado para a pilha
fim_pilha:

; **********************************************************************

PLACE    2000H

; Tabela de vectores de interrupção

tab:     WORD   rot0

; **********************************************************************
; * Programa Principal
; **********************************************************************

PLACE    0

inicio:
  MOV BTE, tab      ; incializa BTE
  MOV R9, INPUT     ; endereço do porto de entrada
  MOV R10, OUTPUT    ; endereço do porto de Ìsada
  MOV SP, fim_pilha
  MOV R5, 1       ; inicializa estado do processo P1
  MOV R6, 1       ; inicializa estado do processo P2
  MOV R4, OFF       ; inicializa controle de RTC
  MOV R8, 0       ; inicializa contador
  MOV R7, OFF       ; inicialmente não permite contagem
  EI0           ; permite interrupções tipo 0
```

```
  EI            ; activa interrupções

ciclo:
  CALL   P1       ; invoca processo P1
  CALL   P2       ; invoca processo P2
  JMP    ciclo    ; repete ciclo

; ***********************************************************************
;* ROTINAS
; ***********************************************************************

P1:
  CMP R5, 1     ; se estado = 1
  JZ  P1_1
  CMP R5, 2     ; se estado = 2
  JZ  P1_2
sai_P1:
  RET         ; sai do processo.


P1_1:
  MOVB   R0, [R9]   ; lê porto de entrada
  BIT R0, 1
  JZ   sai_P1     ; se botão não carregado, sai do processo
  MOV R7, ON      ; permite contagem do display
  MOV R5, 2     ; passa ao estado 2 do P1
  JMP sai_P1

P1_2:
  MOVB   R0, [R9]   ; lê porto de entrada
  BIT R0, 1
  JNZ sai_P1      ; se botão continua carregado, sai do processo
  MOV R7, OFF     ; caso contrário, desliga contagem do display
  MOV R5, 1     ; passa ao estado 1 do P1
  JMP sai_P1
```

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Phasellus eget nisl ut elit porta ullamcorper. Maecenas tincidunt velit quis orci. Sed in dui. Nullam ut mauris eu mi mollis luctus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos.

This inline MATLAB code `for i=1:3, disp('cool'); end;` uses the `\mcode{}` command.[1]

Nullam ut mauris eu mi mollis luctus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Sed cursus cursus velit. Sed a massa. Duis dignissim euismod quam. Nullam euismod metus ut orci.

Listing A.3: Matlab Function

```
for i = 1:3
    if i >= 5 && a ~= b       % literate programming replacement
        disp('cool');         % comment with some LaTeXin it:  πx²
    end
    [:,ind] = max(vec);
    x_last = x(1,end) - 1;
    v(end);
    ylabel('Voltage (µV)');
end
```

[1] MATLAB Works also in footnotes: `for i=1:3, disp('cool'); end;`

Nullam ut mauris eu mi mollis luctus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Sed cursus cursus velit. Sed a massa. Duis dignissim euismod quam. Nullam euismod metus ut orci.

Listing A.4: `function.m`

```
%   Copyright 2010 The MathWorks, Inc.
function ObjTrack(position)
% #codegen
% First, setup the figure
numPts = 300;               % Process and plot 300 samples
figure;hold;grid;           % Prepare plot window
% Main loop
for idx = 1: numPts
    z = position(:,idx);    % Get the input data
    y = kalmanfilter(z);    % Call Kalman filter to estimate the position
    plot_trajectory(z,y);   % Plot the results
end
hold;
end   % of the function
```

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Phasellus eget nisl ut elit porta ullamcorper. Maecenas tincidunt velit quis orci. Sed in dui. Nullam ut mauris eu mi mollis luctus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Sed cursus cursus velit. Sed a massa. Duis dignissim euismod quam. Nullam euismod metus ut orci. Vestibulum erat libero, scelerisque et, porttitor et, varius a, leo.

Listing A.5: HTML with CSS Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Listings Style Test</title>
    <meta charset="UTF-8">
    <style>
      /* CSS Test */
      * {
        padding: 0;
        border: 0;
```

```
      margin: 0;
    }
  </style>
  <link rel="stylesheet" href="css/style.css" />
</head>
<header> hey </header>
<article> this is a article </article>
<body>
  <!-- Paragraphs are fine -->
  <div id="box">
    <p>
      Hello World
    </p>
    <p>Hello World</p>
    <p id="test">Hello World</p>
    <p></p>
  </div>
  <div>Test</div>
  <!-- HTML script is not consistent -->
  <script src="js/benchmark.js"></script>
  <script>
    function createSquare(x, y) {
      // This is a comment.
      var square = document.createElement('div');
      square.style.width = square.style.height = '50px';
      square.style.backgroundColor = 'blue';

      /*
       * This is another comment.
       */
      square.style.position = 'absolute';
      square.style.left = x + 'px';
      square.style.top = y + 'px';

      var body = document.getElementsByTagName('body')[0];
      body.appendChild(square);
    };

```

```
      // Please take a look at +=
      window.addEventListener('mousedown', function(event) {
        // German umlaut test: Berührungspunkt ermitteln
        var x = event.touches[0].pageX;
        var y = event.touches[0].pageY;
        var lookAtThis += 1;
      });
    </script>
  </body>
</html>
```

Nulla dui purus, eleifend vel, consequat non, dictum porta, nulla. Duis ante mi, laoreet ut, commodo eleifend, cursus nec, lorem. Aenean eu est. Etiam imperdiet turpis. Praesent nec augue. Curabitur ligula quam, rutrum id, tempor sed, consequat ac, dui. Vestibulum accumsan eros nec magna. Vestibulum vitae dui. Vestibulum nec ligula et lorem consequat ullamcorper.

Listing A.6: HTML CSS Javascript Code

```

@media only screen and (min-width: 768px) and (max-width: 991px) {

  #main {
    width: 712px;
    padding: 100px 28px 120px;
  }

  /* .mono {
    font-size: 90%;
  } */

  .cssbtn a {
    margin-top: 10px;
    margin-bottom: 10px;
    width: 60px;
    height: 60px;
    font-size: 28px;
    line-height: 62px;
  }
```

Nulla dui purus, eleifend vel, consequat non, dictum porta, nulla. Duis ante mi, laoreet ut, commodo eleifend, cursus nec, lorem. Aenean eu est. Etiam imperdiet turpis. Praesent nec augue. Curabitur ligula quam, rutrum id, tempor sed, consequat ac, dui. Vestibulum accumsan eros nec magna. Vestibulum vitae dui. Vestibulum nec ligula et lorem consequat ullamcorper.

Listing A.7: PYTHON Code

```python
class TelgramRequestHandler(object):
    def handle(self):
        addr = self.client_address[0]          # Client IP-adress
        telgram = self.request.recv(1024)      # Recieve telgram
        print "From: %s, Received: %s" % (addr, telgram)
        return
```

B

A Large Table

Aliquam et nisl vel ligula consectetuer suscipit. Morbi euismod enim eget neque. Donec sagittis massa. Vestibulum quis augue sit amet ipsum laoreet pretium. Nulla facilisi. Duis tincidunt, felis et luctus placerat, ipsum libero vestibulum sem, vitae elementum wisi ipsum a metus. Nulla a enim sed dui hendrerit lobortis. Donec lacinia vulputate magna. Vivamus suscipit lectus at quam. In lectus est, viverra a, ultricies ut, pulvinar vitae, tellus. Donec et lectus et sem rutrum sodales. Morbi cursus. Aliquam a odio. Sed tortor velit, convallis eget, porta interdum, convallis sed, tortor. Phasellus ac libero a lorem auctor mattis. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Nunc auctor bibendum eros. Maecenas porta accumsan mauris. Etiam enim enim, elementum sed, bibendum quis, rhoncus non, metus. Fusce neque dolor, adipiscing sed, consectetuer et, lacinia sit amet, quam. Suspendisse wisi quam, consectetuer in, blandit sed, suscipit eu, eros. Etiam ligula enim, tempor ut, blandit nec, mollis eu, lectus. Nam cursus. Vivamus iaculis. Aenean risus purus, pharetra in, blandit quis, gravida a, turpis. Donec nisl. Aenean eget mi. Fusce mattis est id diam. Phasellus faucibus interdum sapien. Duis quis nunc. Sed enim.

As Table B.1 shows, the results were very satisfactory considering the characteristics of the radio link.

Table B.1: Example table

Benchmark: ANN	**#Layers** (1)	**#Nets** (2)	#Nodes* $(3) = 8 \cdot (1) \cdot (2)$	Critical path $(4) = 4 \cdot (1)$	**Latency (T_{iter})** (5)
A1	**3–1501**	1	**24–12008**	**12–6004**	4
A2	501	1	4008	2004	**2–2000**
A3	10	**2–1024**	**160–81920**	40	60†
A4	10	50	4000	40	**80–1200**
Benchmark: FFT	**FFT size‡** (1)	*#Inputs* $(2) = 2^{(1)}$	*#Nodes** $(3) = 10 \cdot (1) \cdot (2)$	*Critical path* $(4) = 4 \cdot (1)$	**Latency (T_{iter})** (5)
F1	**1–10**	2–1024	**20–102400**	4–40	6–60†
F2	**5**	32	1600	20	**40 – 1500**
Benchmark: Random networks	**#Types** (1)	**#Nodes** (2)	*#Networks* (3)	*Critical path* (4)	**Latency (T_{iter})** (5)
R1	3	10–2000	500	*variable*	(4)
R2	3	50	500	*variable*	$(4) \times [1; \cdots ; 20]$

* *Excluding constant nodes.*
† *Value kept proportional to the critical path:* $(5) = (4) * 1.5$.
‡ *A size of* x *corresponds to a* 2^x *point FFT.*
Values in bold indicate the parameter being varied.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.