CARS - A multi-agent framework to support the decision making in uncertain spatio-temporal real-world applications

Amel Ben Othmane

ECOLE DOCTORALE STIC SCIENCES ET TECHNOLOGIES

DE L'INFORMATION ET DE LA COMMUNICATION

Unité de recherche: I3S/WIMMICS

Thèse de doctorat

Présentée en vue de l'obtention du

grade de docteur en Informatique

de

l'UNIVERSITÉ CÔTE D'AZUR

par

Amel Ben Othmane

CARS - A Multi-agent Framework to Support the Decision Making in Uncertain Spatio-temporal Real-world Applications

Dirigée par *Nhan Le Thanh*

et co-encadrée par *Andrea Tettamanzi* et *Serena Villata*

Soutenue le 12 Octobre 2017 devant le jury composé de :

Andrea TETTAMANZI	Université Côte D'azur	Co-encadreur
Martin ARNAUD	Université de Rennes	Président de jury
Nhan LE THANH	Université Côte D'azur	Directeur de thèse
Viviana MASCARDI	Université de Gênes	Rapporteur
Serena VILLATA	Université Côte D'azur	Co-encadreur

"The only thing certain in life is uncertainty..."

Unknown author

Abstract

Recently, many real-world applications where different entities interact in a dynamic environment, consider the use of agents in their architectures due principally to their autonomy, reactivity and decision-making abilities. Though these systems can be made intelligent, using Artificial Intelligence (AI) techniques, agents still lack of social abilities and have limited knowledge of their environment and in particular when it comes to a dynamic environment. In fact, when operating in the real world, agents need to deal with unexpected events considering both changes in time and space. Moreover, agents must face the uncertainty which pervades real-world scenarios in order to provide an accurate representation of the world.

In this thesis, we introduce and evaluate a formal framework for recommending plans to agents in the decision making process, when they deal with uncertain spatio-temporal information. The agent-based architecture we propose to address this issue, called CARS (Cognitive Agent-based Recommender System), has been designed by extending the well known Belief-Desire-Intention (BDI) architecture to incorporate further capabilities to support reasoning with different types of contextual information, including the social context. Uncertainty on the agent's beliefs, desires and intentions is modeled using possibility theory. To meet the requirements of real-world applications, e.g., traffic and navigation recommendation systems, we define a spatio-temporal representation of the agents' beliefs and intentions. Using such a formal framework, anticipatory reasoning about intentional dynamics can be performed with the aim to recommend an optimal plan to a certain user. Since spatio-temporal data is often considered as incomplete and/or vague, we extended the formal framework with a fuzzy representation of spatio-temporal beliefs and intentions. The framework is evaluated through an Agent Based Simulation (ABS) in a real-world traffic scenario. This ABS allowed us to create a virtual environment to test the impact of the different features of our framework as well as to evaluating the main strengths and weaknesses of the proposed agent architecture.

Résumé

Récemment, plusieurs applications, dans lesquelles différentes entités interagissent dans un environnement dynamique, soulignent l'intérêt de l'utilisation des architectures multi-agents. Ces architectures offrent, dans ce cadre, un certain nombre d'avantages, tels que l'autonomie, la réactivité et la capacité de prise de décision. Elles manquent cependant de capacité sociale et de connaissances sur son environnement, notamment lorsqu'il s'agit d'un environnement dynamique. En effet, quand un agent interagit avec le monde réel, il doit prendre en compte les évènements qui peuvent survenir tout en considérant centaines contraintes telles que le temps et l'espace. En outre, les agents doivent faire face à l'incertitude liée aux applications réelles afin de fournir une représentation fidèle du monde réel.

Dans le cadre de cette thèse, nous proposons un modèle formel de recommandation des plans qui améliore le processus de prise de décision des agents dans un environnement spatio-temporel et incertain. Pour formaliser le comportement cognitifs des agents dans notre système nommé CARS, en anglais "Cognitive Agent-based Recommender System ", nous avons étendu l'architecture BDI qui se base sur le modèle " Croyance-Désir-Intention" pour prendre en compte les différents contextes liés à des applications réelles en particulier le contexte social. Par ailleurs, nous avons également utilisé la théorie possibiliste afin de considérer l'incertitude dans l'état motivationnel d'un agent (c'est à dire ses croyances, désirs, objectifs ou intentions). Pour répondre aux besoins des applications réelles, tels que les systèmes de recommandation relatives au trafic et navigation, nous proposons une représentation spatiotemporelle des croyances et des intentions d'un agent. Cette représentation permettra l'anticipation de certaines intentions, de manière à recommander un plan qui sera optimal pour un utilisateur. Compte tenu l'incomplétude/l'imprécision liée aux données spatiotemporelles, nous avons étendu le modèle proposé pour raisonner avec des croyances et intentions floues. Une évaluation du modèle proposé a été menée en utilisant une simulation multi-agent, dans un scenario réel de circulation routière. Cette simulation a offert un environnement virtuel qui a mis en lumière, après avoir testé les différentes fonctionnalités du modèle, les principaux points forts ainsi que les lacunes liés à l'architecture multi-agents proposée.

Acknowledgements

"When you want something,all the universe conspires in helping you to achieve it."
Paulo Coelho

First I would like to acknowledge persons who have contributed to accomplish this work, my suspervisors Serena Villata and Andrea Tettamanzi. I am very grateful for their support encouragement and invalualable advises. Working with them was a very pleasant experience. I would like also to thank my thesis Advisor Nhan LE thanh for giving me the opportunity to do this thesis and to join the WIMMICS team where I learnt a lot and where I met many great persons. Great thanks also to Michel Buffa and Pavel Arapov with whom I had a great work experience.

There are numerous persons that have supported and inspired me during this long journey. To my friends in the Lab, in particularly Amosse Edouard, Emilie Palagi and Abdoul Macina, I am extremely grateful for your love, support, and encouragement over those years. We went through difficult time together but you were always here to take care of each others. Unfortunately, someone who contibuted a lot in the accomplishment of this work is not here today, but I will never forget you.

Besides my friends I would like to thank all my colleagues in the WIMMICS team. My office co-occupant Alexandre Monin have been a source of support, motivation and learning. And not forgetting our wonderful team assistant Christine Foggia.

I would like to acknowledge the financial support that was provided to me by the French Agency of the Environment and the Energy Management (ADEME) and the Provence Alpes Cote d'Azur (PACA) region. I would like to thank within the ADEME, Valérie Pineau for all her help and Kidness and Gabriel Plassat for accepting to supervise this work.

And last but not least, I thank my husband for being by my side during this journey. And to my son, Jude, I am sorry for making you undergo all this stress.

Contents

List of Figures

List of Tables

Abbreviations

ABS	**A**gent **B**ased **S**imulation
ACL	**A**gent **C**ommunication **L**anguage
AI	**A**rtificial **I**ntelligence
BDI	**B**elief- **D**esire- **I**ntention
BC	**B**elief **C**ontext
CARS	**C**ognitive **A**gent-based **R**ecommender **S**ystem
CBR	**C**ase **B**ased **R**easoning
CBP	**C**ase **B**ased **P**lanning
CB	**C**ontent **B**ased Filtering
CF	**C**ollaborative **F**iltering
CC	**C**ommunication **C**ontext
DC	**D**esire **C**ontext
EV	**E**lectric **V**ehicle
FIPA	**F**oundation for **I**ntelligent **P**hysical **A**gents
GC	**G**oal **C**ontext
GIS	**G**eographic **I**nformation **S**ystem
IC	**I**ntention **C**ontext
KB	**K**nowledge **B**ase
MAS	**M**ulti **A**gents **S**ystems
MCS	**M**ulti **C**ontexts **S**ystems
PC	**P**lanning **C**ontext
POI	**P**oint **O**f **I**nterest
RCC	**R**egion **C**onnection **C**alculus
RS	**R**ecommender **S**ystem

SC	**S**ocial **C**ontext
TMC	**T**raffic **M**essage **C**hannel

Symbols

L_i	Language
A_i	Axioms
Δ_i	Inference rules
Δ_{br}	Bridge rules
π	Possibility distribution
ϕ	A formula
Ω	A set of interpretations
ω	An interpretation
τ	Trust degree
$u(\omega)$	Qualitative utility
$\mu_{\mathcal{A}}$	Membership function
T_M	Minimum T-norm
T_P	Product T-norm
T_W	Łukasiewicz T-norm
$\rightarrow$	Material implication; If ... then
$\wedge$	Logical conjunction; And
$\vee$	Logical disjunction; Or
$\neg$	Logical Negation; Not
$\equiv$	Logical equivalence
$\vdash$	Logical deducibility, provability

To my friend...

You left so early and unexpectedly...

Chapter 1

Introduction

Nowadays, recommender systems must cope with the increasing demand of complexity real-world scenarios ask for, e.g., a recommendation application for recommending routes in a traffic scenario should deal with different contextual information like information about the user location and other non-logical components of human behavior like desires, beliefs or emotions. Although traditional recommendation techniques (i.e., content-based [4], collaborative filtering [5] or hybrid ones [6]) have been enhanced to meet users' requirements by including, for instance, Semantic Web techniques or context-aware information, they fail to give personalized recommendation when the targets are not simple e-commerce items but instead further complex plans.

For this reason, agents and Multi-Agent systems are considered as suitable alternatives for modeling and simulating this kind of real-world scenarios, where different entities interact in a *dynamic* and *uncertain* environment. In particular, one of the most popular agent architectures, the Belief-Desire-Intention (BDI) model [7], seems to be particularly suitable to the task. Under this model, the mental state of the agent is composed by sets of *beliefs*, *desires* and *intentions* that consist of informational, motivational, and deliberative states, respectively.

Recently, the Artificial Intelligence (AI) community is putting much effort on the investigation and evaluation of recommender systems based on intelligent agents. Such a kind of systems has been applied so far in different fields such as healthcare [8], tourism [1], financial applications [9], and traffic and transportation [10]. A complete taxonomy of recommender agents can be found in [11].

The advantage of such a kind of recommender systems is that of encoding users' beliefs and goals in the system to return a recommendation as close as possible to the users' needs, with the possibility to include additional information like the confidence in the source. Nevertheless many research challenges remain open in this area.

First, several of the above application scenarios require to formalize the knowledge about the time and the location in which the action is taking place. These pieces of information often need to be considered together, as in the case of the traffic scenario where a traffic jam is identified by its location and the time it is occurring during the day, and require to encode a certain degree of vagueness as well.

Second, agents have to represent user's beliefs, desires or intentions in such a way to encode their imprecision or vagueness, as it holds for human-based reasoning. For instance, a user may provide to the recommender system a vague goal such as "I want to be at home around 9 am".

1.1 Motivations

A few illustrative examples are presented here to demonstrate the need of Multi-Agent systems in engineering applications, and to motivate the problems considered in this thesis. The first one is in the health-care domain, and the second one is in the traffic domain.

Bob, a 40 year-old adult, wants to get back to a regular physical activity (pa). Bob believes that a regular physical activity reduces the risk of developing a non-insulin dependant diabetes mellitus (rd). Mechanisms that are responsible for this are weight reduction (wr), increased insulin sensitivity, and improved glucose metabolism. Due to his busy schedule (bs), Bob is available only on weekends (av). Hence, he would be happy if he could do his exercises only on weekends (w). Bob prefers also not to change his eating habits (eh). Besides all the aforementioned preferences, Bob should take into account his medical concerns (c) and certainly refers to a health-care provider for monitoring.

This scenario exposes the following problem: *how can we help Bob to select the best plan to achieve his goal based on his current preferences and restrictions?* This problem raises different challenges. First, the proposed solution should take into account Bob's preferences and restrictions (e.g., medical and physical concerns) in the recommendation process. Second, information about the environment in which Bob acts, and people that might be in relationship with him, may have impact in his decision-making process. Third, the system should be able to keep a trace of Bob's activities in order to adapt the recommendation according to his progress. Finally, the information or data about Bob's activities is distributed geographically and temporally.

The same problems are raised in the traffic scenario with some particularities related to the traffic field. *Suppose that Bob uses an electric car, and needs to reach a public electric charging point. Like any road user, Bob relies on a navigation system to determine the nearby charging points before his journey. Knowing the time needed to get to the charging point and the battery life, Bob can decide where and when to leave.*

This scenario exposes some further problems related to classical navigation recommender systems, that can handle simple scenarios where the user only needs to reach a destination. Nevertheless, in cases when some events need to be handled (i.e., battery life, accidents, ...) or when users have more sophisticated requirements (e.g., choosing a route with a nice landscape), these systems lack from the expertise and autonomy points of view. Besides, in such scenario, it is interesting to exploit the community network (electric cars users network or route users network) in order to anticipate some events and hence enhance the quality of the recommendation to get the optimal route.

1.2 Research questions

In this thesis, we answer the research questions raised earlier on in this Section, and motivated by the two scenarios described in Section 1.1:

— how to define a recommender system able to deal with the flexibility, complexity and dynamics required for real-world applications?

— how to represent and reason about fuzzy spatial-temporal knowledge to provide useful recommendations?

1.3 Main Contributions

To address these research questions, in this thesis we propose:

(i) A multi-context recommender system based on the BDI architecture, called **CARS** (**C**ognitive **A**gent-based **R**ecommender **S**ystem). The proposed framework aims at recommending a plan for a user taking into account different contexts. For this purpose, we combined two different approaches to define the different components of our framework :*(1)* an implementation of a full-fledged possibilistic BDI model of agency which integrates goal generation, inspired from da Costa Pereira and Tettamanzi [12, 13], and *(2)* multi-context systems applied to the BDI architecture, inspired from Parsons *et al.* [14], to define the different theories and contexts that are put together to define the whole framework. We also extend the BDI model with extra contexts to enrich agents with social and functional capabilities.

(ii) An agent-based simulation study to evaluate CARS in the Netlogo Platform.[1] To evaluate the performance of the system, we use two different strategies, namely the *solitary agent* strategy, where agents operate individually without communicating with the other agents in the Multi-Agent System (MAS), and the *social agent* strategy, where agents consider information coming from the other agents in the MAS. We consider in this simulation agents with random distribution (random beliefs and desires, and random positions in the environment).

(iii) An extension of CARS with fuzzy spatio-temporal information. Based on the extension principle of fuzzy set theory [15], we define fuzzy Allen's intervals [16] to model temporal knowledge, while fuzzy topological relations are defined in terms of Region Connection Calculus (RCC) [17] where regions are represented as fuzzy sets. These two components, namely spatial and temporal information, are combined together based on the assumption that the degree to which a spatio-temporal belief is true is the *minimum* between the confidence degrees of the spatial belief and the temporal one, respectively. Spatio-temporal knowledge is thus exploited by agents to update their beliefs following the other agents' recommendations, with the aim to reach their goals.

1. https://ccl.northwestern.edu/netlogo/

(iv) An empirical evaluation of the extended version of CARS in a simulated environment using the NetLogo Platform enhanced with the GIS extension to show the advantages of the proposed agent-based recommender system. We consider a traffic scenario where the goal of the agents is to reach a certain Point Of Interest (POI) as fast as possible. Agents communicate about possible accidents and traffic jams taking place around a certain time and in a certain geographical zone, and suggest alternative routes to help the other agents to reach their destinations. We consider the same agent strategies used to evaluate the first version of the CARS system.

1.4 Outline of the thesis

The remainder of this thesis is structured as follows:

— Chapter 2 outlines some basics indispensable to understand the system design and experiments presented in the thesis. It comprises background material and establishes the mathematical notation that will be used throughout the thesis. Background is presented in four main areas: Agents and Multi-Agents Systems, Uncertainty Reasoning, Spatial Reasoning and Temporal reasoning.

— Chapter 3 presents a literature survey on agent-based recommender systems in two different areas: the traffic and tourism domains. It also gives an overview of a specific type of agent-based recommender systems, namely, BDI-based recommender systems. We also reviewed approaches about temporal and spatial reasoning applied to recommender systems. This chapter provides us with a state-of-the-art description of agent-based recommender systems.

— Chapter 4 provides the contribution of the thesis. The first part introduces **CARS**, the multi-context BDI recommender framework, highlighting the main features of the system and its behavior. The behavior is described through the specification of the different contexts and the different rules used to rely together all those contexts. An empirical evaluation of the proposed framework using Multi-Agent simulation is also presented and results are discussed. In the second part of this chapter, we introduce the spatio-temporal version of *CARS*, an extension of the multi-context

BDI recommender framework presented in the first part, with fuzzy spatio-temporal reasoning. In this section, we formally define the spatio-temporal fuzzy representation of the agents' beliefs as well as their update mechanism. An evaluation of this extension in the traffic domain using Netlogo with a GIS is discussed, to show the usefulness of the proposed agent-based recommender system.

— Chapter 5 concludes the thesis summarizing its main contributions, and listing some open issues left as future work. A list of the publications related with the thesis is included.

Chapter 2

Background

In this Chapter, we provide some prerequisites relevant to the design and development of our agent-based formal framework by surveying the most important methods and formalisms we rely upon.

An important prerequisite to build a Multi-Agent system is the ability to identify the appropriate software/hardware structure. For this reason, we briefly report about the different agent architectures in the literature, and then we concentrate on a particular architecture: the Belief-Desire-Intention model. We are interested in a specific method for defining architectures for logic-based agents, i.e., the use of multi-context systems which allows distinct theoretical components to be defined and interrelated. We provide some examples of BDI agent specifications using multi-context systems.

Since uncertainty is unavoidable in everyday reasoning, we present different ways to handle it in real-world applications with a particular focus on requirements for reasoning under uncertainty with spatial and temporal features.

2.1 Agents and Multi-Agent systems

There is no universally accepted definition of the term of "agent". However, even if researchers were not able to agree on a universal consensus, there are many accepted definitions within the Artificial Intelligence community. One of the most well-known definitions of the concept of agent was introduced by Jacques Ferber [18]. According to this definition, an agent is a physical entity:

— which is able to act in an environment;
— which can communicate directly with other agents;
— which is driven by a set of tendencies (in the form of individuals objectives)
— which possesses its own resources;
— which is capable of perceiving its environment (in a limited manner)
— which has skills and offers services
— whose behavior tends to satisfy objectives, while taking the resources and skills into account, and as a function of its perception, representations, and the communications it receives.

Wooldridge and Jennings' definition distinguishes between and agent and an intelligent agent, which is further required to be autonomous, reactive, proactive and social [19]:

— *autonomy*: agents are independent and make their own decisions without direct intervention of other agents or humans and agents have control over their actions and their internal state.
— *reactivity*: agents need to be reactive, responding in a timely manner to changes in their environment.
— *pro-activity*: an agent pursues goals over time and takes the initiative when it considers it appropriate.
— *sociability*: agents very often need to interact with other agents to complete their tasks and help others to achieve their goals.

The Wooldridge and Jennings definition, in addition to spelling out autonomy, sensing and acting, allows for a broad, but finite, range of environments. They further add a communications requirement. That's why in this thesis we will consider the latter.

2.1.1 Agents architectures and theories

As defined by Maes in [20], an agent architecture is :"*a particular methodology for building [agents]. It specifies how ... the agent can be decomposed into the construction of a set of component modules and how these modules should be made to interact. The total set of modules and their interactions has to provide an answer to the question of how the sensor data and the current internal state of* *the agent determine the actions ... and future internal state of the agent. An architecture encompasses techniques and algorithms that support this methodology.*"

Different architectures encapsulate different approaches to a rational decision making and we are going to overview in the next sub-sections some of the well-known agents' architectures based on the classification of Wooldridge and Jenning's in [19].

2.1.1.1 Deliberative agents

A deliberative or a logic-based agent architecture is one of the earliest agent architectures that rest on the physical-symbol systems hypothesis [21]. An agent in such architecture contains an explicitly represented, symbolic model of the world, in which decisions (for example about what actions to perform) are made via logical (or at least pseudo-logical) reasoning, based on pattern matching and symbolic manipulation.The syntactical manipulation of the symbolic representation is the process of logical deduction or theorem proving.

Earlier attempts to use deliberative reasoning led to STRIPS (Stanford Research Institute Problem Solver) [22]. However, it soon became obvious that STRIPS concept needed further improvement. In fact, it was unable to effectively solve problems of even moderate complexity. More successful attempts using this architecture include the Belief-Desire-Intention (BDI) [7] architecture which is considered as a logic-based architecture. However, due to its popularity and wide adoption, the discussion on this particular architecture is detailed in Section 2.1.1.4.

Two core issues within logic-based agents were recognized which resulted in developing a reactive architecture:

— The transduction problem: it is difficult and time consuming to translate all of the needed information into the symbolic representation, especially if the environment is changing rapidly.

— A representation/reasoning problem: It is very difficult or sometimes impossible to put down all the rules for the situation that will be encountered by the agent in a complex environment since the deduction process is based on set of inference rules.

2.1.1.2 Reactive agents

Unlike a deliberative agent, which possesses an internal image of the external environment thanks to the symbolic representation it maintains to reach its goal, a reactive agent is able to reach its goal only by reacting reflexively on external stimuli.

Woodridge and Jennings [23] define the reactive architecture to be the opposite of the deliberative by defining it to be "one that does not include any kind of central symbolic world model, and does not use complex symbolic reasoning". Brook's subsumption architecture [24] is one of the most known purely reactive architectures. Instead of modelling aspects of human intelligence via symbol manipulation, this approach is aimed at real-time interaction.

2.1.1.3 Hybrid agents

Many researchers have suggested that neither a completely deliberative nor completely reactive approach is suitable for building agents. An obvious approach is to build an agent out of two (or more) subsystems composed of a deliberative one that develops plans and makes decisions using a symbolic reasoning and a reactive one capable of reacting to events without complex reasoning. Subsystems are decomposed into a hierarchy of interacting layers to deal with reactive and pro-active behaviours respectively.

Layering is a powerful means for structuring functionalities and control, and thus is a valuable tool for system design supporting several desired properties such as reactivity, deliberation, cooperation and adaptability. The main idea is to structure the functionalities of an agent into two or more hierarchically organized layers that interact with each other to achieve coherent behaviour of the agent as a whole. The *Touring Machine* [25] introduced by Ferguson is an example of a layered control architecture for autonomous, mobile agents performing constrained navigation tasks in a dynamic environment.

2.1.1.4 The Belief-Desire-Intention Architecture

The origin of this architecture lies in the theory of human practical reasoning introduced by the philosopher Michael Bratman [26]. Bratman defined practical reasoning as "*a matter of weighing conflicting considerations for and against competing options where the relevant considerations are provided by what the agent desires and what the agent believes*". Practical reasoning is composed of two important processes: deciding what state of affairs we want to achieve known as *deliberation*, and how we are going to achieve these goals called *means-ends reasoning*. In the BDI architecture, an agent consists of three logic components referred as mental states namely beliefs, desires and intentions. Beliefs encode the agent's understanding of the environment, desires are those states of affairs that an agent would like to accomplish while intention is more concerned with agent's committing to obtain this state of affairs otherwise called goal. To gain an understanding of the BDI model, it is worth considering a simple example of practical reasoning. For example, if we *desire* to be an academic, then you would expect us to apply for various PhD programs in order to achieve this goal. Of course if our application is accepted then we should commit to this objective and devote time and effort to achieve it. By this, we mean that we would carry out some course of action that we *believed* would best satisfy our objective. So these actions would be our *intentions* and we will commit to act upon until they are achieved or dropped because we believe they will never be achieved.

Many approaches tried to formalize such mental attitudes (e.g., [27], [7], [28] and [29]). Rao and Georgeff [7] formalized the BDI model, including the definition of the underlying logic, the description of belief, desire and intentions as modal operators, the definition of a possible worlds semantics for these operators, and an axiomatisation defining the interrelationship and properties of the BDI-operators.

The BDI model is attractive for several reasons. First, it is intuitive — we all recognize the processes of deciding what to do and then how to do it, and we all have an informal understanding of the notions of belief, desire, and intention. Second, it gives us a clear functional decomposition, which indicates what sorts of subsystems might be required to build an agent. However, the main difficulty is how to achieve a good balance between proactive (goal-directed) and reactive (event driven) behaviors.

There are a number of implementations of BDI agents. The most popular ones are Rao and Georgeff BDI Logics, the Procedural Reasoning System (PRS) and its more recent incarnation, the Distributed Multi-Agent Reasoning System (dMARS). Another implementation inspired from the previous ones is AgentSpeak(L).

2.2 Multi-context Systems

Multi-Context Systems (MCSs) were introduced in [30] to address the need for a general framework that integrates knowledge bases expressed in heterogeneous formalisms. Intuitively, instead of designing a unifying language to which other languages could be translated, in an MCS the different formalisms and knowledge bases are considered as modules, and means are provided to model the flow of information between them. More specifically, MCSs are a formalization of simultaneous reasoning in multiple contexts. Different contexts are inter-linked by bridge rules which allow for a partial mapping between formulae/concepts/information in different contexts.

Following the formalization proposed in [31], a multi-context system (MCS) (or a *Multi-language System*) consists of a collection of contexts (or units), each of which contains knowledge represented in some logic, and a set of bridge rules. In addition to the logic in each context, bridge rules are used to interconnect the contexts. Let I be the set of context names, a MCS is formalized as $\{C_i\}_{i \in I}, \Delta_{br}$, where:

— For each $i \in I$, $C_i = \langle L_i, A_i, \Delta_i \rangle$ is an axiomatic formal system where L_i, A_i and Δ_i are the language, axioms, and inference rules respectively. They define the logic for context C_i whose basic behavior is constrained by the axioms.

— Δ_{br} is a set of bridge rules.

Bridge rules can be seen as rules of inference which relate formulae in different contexts. A bridge rule is typically written as follows:

$$\frac{C_1 : \phi_1, ..., C_n : \phi_n}{C_x : \phi_x}$$

and can be read as follows: if formulae $\phi_1, ..., \phi_n$ hold in their respective contexts $C_1, ..., C_n$, then the formula ϕ_x is true in the context C_x.

Using multi-context systems for specifying and modelling agent architectures turns out to be suitable for multiple reasons: (i) from a software engineering perspective they support modular decomposition and encapsulation; and (ii) from a logical modelling perspective they provide an efficient means of specifying and executing complex logics. This considerably increases the representation power of logical agents, and at the same time, simplifies their conceptualization. Several works have appeared where MCS are used to specify agents.

2.2.1 Multi-context Agents

An agent can be viewed as a multi-context system in which each of the architecture's blocks is represented as a separate unit, an encapsulated set of axioms and an associated deductive mechanism, whose interrelationships are precisely defined via bridge rules, inference rules connecting units. Using a multi-context approach, a multi-context agent architecture consists of four basic types of component as defined by Parsons in [14]:

— *Units:* Structural entities representing the main components of the architecture.

— *Logic:* Declarative languages, each with a set of axioms and a number of rules of inference. Each unit has a single logic associated with it.

— *Theories:* set of formulae written in the logic associated with a unit.

— *Bridge rules:* Rules of inference which relate formulae in different units

Units represent the various components of the architecture. They contain the mass of an agent's problem solving knowledge, and this knowledge is encoded in the specific theory that the unit encapsulates. For example, a BDI agent may have units which represent theories of beliefs, desires and intentions.

2.3 Reasoning under uncertainty

2.3.1 Fuzzy Sets

Fuzzy set theory was introduced by Zadeh in the 1960s (see [32] for more details) and deals with sets or categories whose boundaries are 'fuzzy'. In other words, a fuzzy set is a set of objects whose membership to the set takes a value between zero and one. Each fuzzy object can have partial or multiple memberships. Let $\mathcal{X}$ be a classical set of objects, called the *the universe*, whose elements are denoted x. A fuzzy set A in $\mathcal{X}$ is mathematically characterized by a membership function $\mu_A(x)$ which associates with each x in X a real number in the interval $[0, 1]$, with the membership value at x representing the "degree of membership" of x inA.

Membership in a classical subset of $\mathcal{A}$ of $\mathcal{X}$ is defined by the characteristic function $\mu_{\mathcal{A}}$ from $\mathcal{A}$ to $\{0, 1\}$ such that:

$$\mu_{\mathcal{A}}(x) = \begin{cases} 1 & \text{iff} \quad x \in \mathcal{A} \\ 0 & \text{iff} \quad x \notin \mathcal{A} \end{cases}$$

Clearly, $\mathcal{A}$ is a subset of $\mathcal{X}$ that has no sharp boundary and is characterized by a set of pairs $\mathcal{A} = \{(x, \mu_{\mathcal{A}}(x)), x \in \mathcal{X}\}$. When X is a finite set $\{x_1, ..., x_n\}$, a fuzzy set is expressed as:

$$\mathcal{A} = \sum_{i=1}^{n} \mu_{\mathcal{A}}(x_i)/x_i$$

When x is not finite, we write:

$$\mathcal{A} = \int_{\mathcal{X}} \mu_{\mathcal{A}}(x)/x$$

2.3.1.1 The extension principle

The extension principle, introduced by Zadeh [15], provides a way to extend non-fuzzy mathematical concepts in order to deal with fuzzy quantities. In general the extension principal is defined by the following equation:

$$\mu_{A*B}(z) = \sup_{z=x*y} \min\{\mu_A(x), \mu_B(y)\} \tag{2.1}$$

where $\forall x, y \in \mathcal{X}$, $\mu_A(x) \in [0,1]$ and $\mu_B(y) \in [0,1]$ are membership functions defining the degree of belonging of the elements of $\mathcal{X}$ to the fuzzy subsets A and B, respectively. The symbol $*$ denotes any crisp operator. Then a few consequences of applying fuzzy function to some logical operator are the following :

$$\mu_{X \wedge Y} = \min(\mu_X, \mu_Y)$$
$$\mu_{X \vee Y} = \max(\mu_X, \mu_Y)$$
$$\mu_{\neg X} = 1 - \mu_X$$

The union $\cup$ and intersection $\cap$ of ordinary subsets of $\mathcal{X}$ can be extended by the following formula proposed by Zadeh:

$$\forall x \in \mathcal{X},\ \mu_{A \cup B} = \max\left(\mu_A(x),\ \mu_B(x)\right) \tag{2.2}$$

$$\forall x \in \mathcal{X},\ \mu_{A \cap B} = \min\left(\mu_A(x),\ \mu_B(x)\right) \tag{2.3}$$

where $\mu_{A \cup B}$ and $\mu_{A \cap B}$ are respectively the membership functions of $A \cup B$ and $A \cap B$.

2.3.1.2 T-norms and T-conorms

T-norms and T-conorms [33, 34] are used to calculate the membership values of intersection and union of fuzzy sets, respectively. A T-norm is a binary operation $T : [0, 1]^2 \to [0, 1]$ satisfying the following axioms for all $x, y, z \in [0, 1]$:

(i) $T(x, y) = T(y, x)$ (commutativity),

(ii) $T(x, y) \leq T(x, z), if\ y \leq z$ (monotonicity),

(iii) $T(x, T(y, z)) = T(T(x, y), z)$ (associativity),

(iv) $T(x, 1) = x$

Some common T-norms (and respectively their corresponding T-conorms) are the minimum $T_M(S_M)$, the product $T_P(S_P)$ and the Łukasiewicz $T_W(S_W)$ defined as:

$$T_M(x, y) = \min(x, y),\ S_M(x, y) = \max(x, y)$$

$$T_P(x, y) = x.y,\ S_M(x, y) = x + y + xy$$

$$T_W(x, y) = \max(0, x + y - 1),\ S_W(x, y) = \min(1, x + y)$$

Implicators generalize the logical implication to the unit interval and are defined by $I_S(x, y) = S(1 - x, y)$ for x and y in [0, 1]. For example the implicator corresponding to S_M is defined by $I_{S_M}(x, y) = \max(1 - x, y)$.

2.3.2 Possibility Theory

Possibility theory is an uncertainty theory dedicated to handle incomplete information. It was introduced by [35] as an extension to fuzzy sets. Possibility theory differs from probability theory by the use of dual set functions (possibility and necessity measures) instead of only one. A possibility distribution assigns to each element ω in a set Ω of interpretations a degree of possibility $\pi(\omega) \in [0, 1]$ of being the right description of a state of affairs. It represents a flexible restriction on what is the actual state with the following conventions:

— $\pi(\omega) = 0$ means that state ω is rejected as impossible;

— $\pi(\omega) = 1$ means that state ω is totally possible (plausible).

2.4 Agent-based Modelling and Simulation

2.4.1 Agent-based simulation

Agent-Based Simulation (ABS) is a computational technique for modelling complex systems composed of interacting autonomous individuals (i.e., agents) in a network.

The advantage of simulation compared to other research methods is primarily the fact that the designer is in control of any parameter to adapt to a specific problem. This allows for both, normative and descriptive studies. A well-designed simulation system can help to understand and explain real world systems, and to describe certain observed phenomena by comparing different simulation settings. Agent-Based Simulation provides some additional advantages. According to [36], ABS allows for modeling complex behavior of an agent without restrictions on the complexity of its reasoning, on the sophistication of its internal structure, or on its interaction abilities. Bonabeau [37] summarizes the benefits of ABS over other modeling techniques as follows:

— ABS captures emergent phenomena,

— ABS provides a natural description of a system,

— ABS is flexible.

ABS is particularly suitable in the social context where a large number of human agents interact and co-operate for common goals. Therefore, we next focus on a particular agent architecture which is the BDI architecture.There are several simulation frameworks supporting the creation of agents defined using these three components, and a huge number of systems extending them to provide additional human reasoning capabilities. In [38] for example, the authors propose a simulation of military commanders in land operation scenarios using the Jack framework. Cecconi and Parisi [39] propose the use of simulation to evaluate various survival strategies of individuals in a social group. In this simulation, agents adopts two strategies: the individual survival strategy and the social survival strategy. In [40], the authors present a crowd simulation for emergency response where agents are implemented with an extended BDI architecture, which includes an emotional component and a real-time planner. One can find further examples in [41] and [42] which give an overview on ABS applications including those using BDI agents.

However, to the best of our knowledge, there is no application of ABS in recommender systems combined with the BDI architecture. Later in this paper, we will evaluate different strategies of a BDI recommender system using ABS.

2.4.2 Platforms for agent-based simulation

There are several tools that are designed for ABS, as shown in Table 2.1. The focus here will be on general purpose and freely available ones. A more extensive study can be found in [43, 44]. Swarm [45] was the first ABS software development environment launched in 1994 at the Santa Fe Institute. It was designed as a general language and toolbox for agent-based modeling and simulation, intended to have a widespread use across scientific domains. It was written originally in Objective C which make it in practice not easy to use. In fact, it is necessary to have experience in Objective C to be familiar with Swarm platform. The Repast (REcursive Porous Agent Simulation Toolkit) toolkit [46] had the initial objective to implement Swarm in Java. However, it significantly diverged from Swarm. It focuses on social behavior, in the social science domain, and offers support tools for social networks. There are three implementations of Repast: Repast for Java (Repast J), Repast for the Microsoft.Net framework (Repast.Net), and Repast for Python Scripting (Repast Py). Mason [47] was designed as a faster alternative to Repast. Its main objective, compared to Repast, is clearly to maximize execution speed with a focus on computationally demanding models. However, it is not an easy to learn toolkit, as it requires significant Java Knowledge.

TABLE 2.1: A comparison of agent-based simulation platforms

Criteria/Platform	Mason [1]	Repast [2]	NetLogo [3]	Swarm [4]
Language	Java	Java, C++ Python	Logo	Objective C Java
Execution Speed	Moderate	Fast	Moderate	Slow
Documentation	Good	Poor	Large	Good
Learning facilities	Moderate	Moderate	Moderate	Moderate
Primary domain	Social sciences	Social sciences	General purpose	General purpose

[1] http://cs.gmu.edu/ eclab/projects/mason/re
[2] http://repast.sourceforge.net/
[3] https://ccl.northwestern.edu/netlogo/
[4] http://www.swarm.org/

NetLogo [48] is a free and open source agent-based simulation environment that uses a modified version of the Logo programming language, built-in graphical interfaces, and comprehensive documentation. NetLogo provides a graphical environment to create programs that control graphic turtles that reside in a world of patches, which are monitored by an observer. Links are also available to connect turtles to form networks. NetLogo is highly recommended [44], even for prototyping complex models. Each agent in Netlogo:

— perceives its environment and acts upon it,

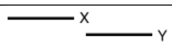
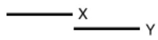
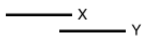
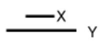
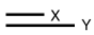
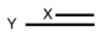
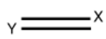
— carries its own thread of control, and

— is autonomous.

2.5 Allen's Intervals Algebra

Allen's Interval approach [16] is an algebra of binary relations on intervals for representing qualitative temporal information and addresses the problem of reasoning about such information. Allen's approach is based on the notion of time intervals and binary relations on them. A time interval X is an ordered pair $\langle X^-, X^+\rangle$ such that $X^- < X^+$, where X^- and X^+ are interpreted respectively as the starting and ending points of the interval. Allen's introduces thirteen basic interval relations illustrated in Table 2.2: $\prec$ (before), m (meets), o (overlaps), d (during), s (starts), f (finishes), their converse relations ($\succ$, m_i, o_i, d_i, s_i, f_i), and $=$ (equal), where each basic relation can be defined in terms of its endpoint relations. For example, the interval relationship $X\,d\,Y$ (interval X during the interval Y) can be expressed as $(X^- > Y^-) \wedge (X^+ < Y^+)$. We refer the interested reader to [16] for a more detailed discussion about Allen's intervals.

TABLE 2.2: Allen's thirteen time relations.

Relation	**Converse**	**Pictorial Example**	Endpoint Relations
$X \prec Y$	$X \succ Y$		$X^+ < Y^-$
XmY	Xm_iY		$X^+ = Y^-$
XoY	Xo_iY		$X^- < Y^-$, $X^+ > Y^-$, $X^+ < Y^+$
XdY	Xd_iY		$X^- > Y^-$, $X^+ < Y^+$
XsY	Xs_iY		$X^- = Y^-$, $X^+ < Y^+$
XfY	Xf_iY		$X^- < Y^-$, $X^+ = Y^+$
$X = Y$	$X = Y$		$X^- = Y^-$, $X^+ = Y^+$

2.6 Region Connection Calculus

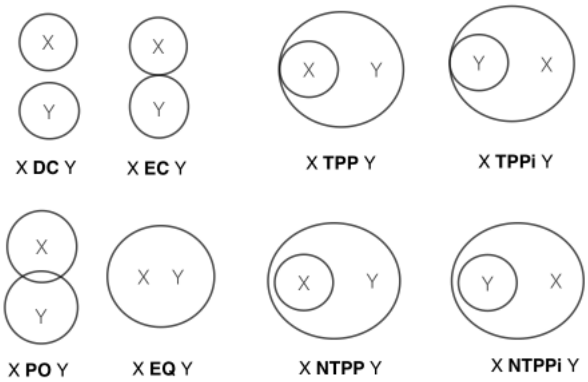


FIGURE 2.1: The main RCC-8 relations.

One of the most important formalisms for topological relationships is the Region Connection Calculus (RCC) [17]. The RCC is an axiomatization of certain spatial concepts and relations in first order logic. The basic theory assumes just one primitive dyadic relation: $C(x, y)$ read as "x connects with y". RCC has eight basic relations (illustrated in Figure 2.1): DC (DisConnected), EC (Externally Connected), PO (Partial Overlap), EQ (EQual), TPP (Tangential Proper Part), NTPP (Non Tangential Proper Part) and their converse relations TPPi (TPP inverse) and NTPPi (NTPP inverse).

For further details about RCC, we refer the reader to [17].

TABLE 2.3: Definition of spatial relations entailed in the RCC. U is the universe of all regions. x and y are variables denoting arbitrary elements of U, i.e. regions

Name	**Relation**	**Definition**
Disconnected	$DC(x,y)$	$\neg C(x,y)$
Part	$P(x,y)$	$\forall z \in U, C(z,x) \rightarrow C(z,y)$
Proper Part	$PP(x,y)$	$P(x,y) \wedge \neg P(y,x)$
Equals	$EQ(x,y)$	$P(x,y) \wedge P(y,x)$
Overlaps	$O(x,y)$	$\exists z \in U, P(z,x) \wedge P(z,y)$
Discrete	$DR(x,y)$	$\neg O(x,y)$
Partially Overlaps	$PO(x,y)$	$O(x,y) \wedge \neg P(x,y) \wedge \neg P(y,x)$
Externally connects	$EC(x,y)$	$C(x,y) \wedge \neg O(x,y)$
Tangential Proper Part	TPP	$PP(x,y) \wedge (\exists z \in U, EC(z,x) \wedge EC(z,y))$
Non-Tangantial Proper Part	$NTPP(x,y)$	$PP(x,y) \wedge \neg(\exists z \in U, EC(z,x) \wedge EC(z,y))$

Chapter 3

A Cognitive Agent-based Recommender System (CARS)

In this Chapter, the **CARS** framework is introduced. The design of such system is motivated by two main goals:

1. to enhance recommender systems with reasoning and autonomous decision-making abilities in order to deal with the complexity, flexibility and dynamics required in real-world applications.
2. to provide personalized and useful recommendation to users by handling uncertain spatio-temporal reasoning, necessary in such real world applications.

The main aim of the proposed framework (as illustrated in Figure 3.1 inspired from the scenario introduced in Chapter 1 (Section 1.1) is to recommend to users a list of activities based on their preferences/restrictions, and their own beliefs.

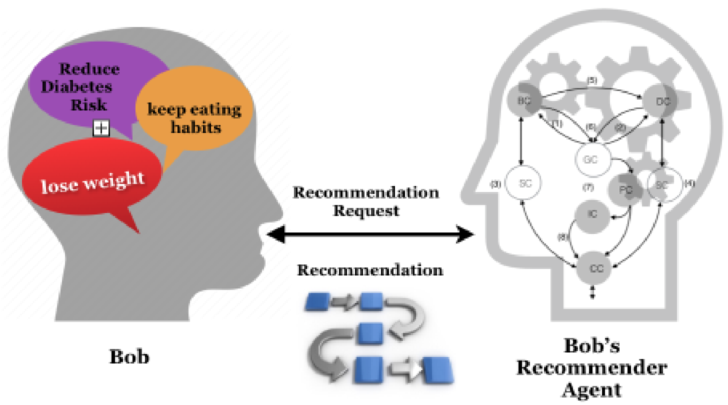


FIGURE 3.1: A use case of CARS Framework

3.1 A Multi-context BDI recommender Framework

The BDI agent architecture we are proposing in this thesis extends Rao and Georgeff's well-known BDI architecture [7]. We define a BDI agent as a multi-context system being inspired by the work of [14]. Following this approach, our BDI agent model, visualized in Figure 3.2, is defined as follows:

$$Ag = (\{BC, DC, GC, SC, PC, IC, CC\}, \Delta_{br})$$

where BC, DC, GC represent respectively the Belief Context, the Desire Context and the Goal Context which model an agent mental attitude. PC, IC and CC are functional contexts that represent respectively the Planning Context, the Intention Context, and the Communication Context. SC is for the Social Context, and it models social influence between agents.

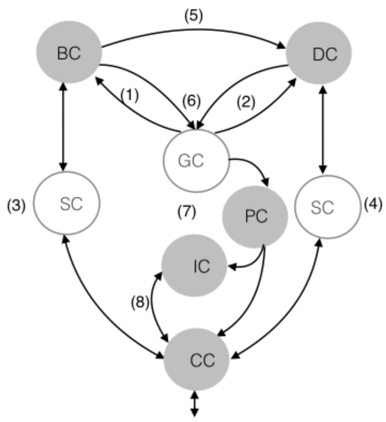


FIGURE 3.2: The extended Multi-context BDI agent model.

In order to reason about beliefs, desires, goals and social contexts we follow the approach developed by da Costa Pereira and Tettamanzi [12, 13], where they adopt a classical propositional language for the representation of beliefs, desires, and intentions, and possibility theory to deal with uncertainty.

Let $\mathcal{A}$ be a finite set of atomic propositions, and $\mathcal{L}$ be the propositional language such that $\mathcal{A} \cup \{\top, \bot\} \subseteq \mathcal{L}$ and $\forall \phi, \psi \in \mathcal{L}, \neg\phi \in \mathcal{L}, \phi \vee \psi \in \mathcal{L}, \phi \wedge \psi \in \mathcal{L}$. These propositions can contain temporal elements, but dealing with these elements is left as future work. As in [12], $\mathcal{L}$ is extended, and we will denote with $\Omega = \{0, 1\}^{\mathcal{A}}$ the set of all interpretations on $\mathcal{A}$. An interpretation $\omega \in \Omega$ is a function ω :

$\mathcal{A} \to \{0,1\}$ assigning a truth value p^ω to every atomic proposition $p \in \mathcal{A}$ and, by extension, a truth value ϕ^ω to all formulae $\phi \in \mathcal{L}$. $[\phi]$ denotes the set of all interpretations satisfying ϕ, i.e., $[\phi] = \{\omega \in \Omega : \omega \vDash \phi\}$.

In the Planning and Intentions contexts, we propose an ontological representation for plans and intentions to provide the agents with a computer-interpretable description of the services they offer, and the information they have access to. In the following subsections, we will outline the different theories defined for each context of our multi-context agent model.

3.1.1 Belief Context

An agent's belief represents the information about the world as well as information coming from other agents. An agent may update its beliefs by observing the world and by receiving messages from other agents.

3.1.1.1 The *BC* language and semantics

In order to represent beliefs, we use the classical propositional language with additional connectives, following [12]. We introduce also a fuzzy operator B over this logic to represent agent's beliefs. The belief of an agent is then represented as a possibility distribution π. A possibility distribution π can represent a complete preorder on the set of possible interpretations $\omega \in \Omega$. This is the reason why, intuitively, at a semantic level, a possibility distribution can represent the available knowledge (or beliefs) of an agent. When representing knowledge, $\pi(\omega)$ acts as a restriction on possible interpretations and represents the degree of compatibility of the interpretation ω with the available knowledge about the real world. $\pi(\omega) = 1$ means that is totally possible for ω to be the real world. As in [12], a graded belief is regarded as a necessity degree induced by a normalized possibility distribution π on the possible worlds ω. The degree to which an agent believes that a formula Φ is true is given by:

$$B(\phi) = N([\phi]) = 1 - \max_{\omega \nvDash \phi}\{\pi(\omega)\} \tag{3.1}$$

An agent's belief can change over time because new information arrives from the environment or from other agents. A belief change operator is proposed in [12], which allows to update the possibility distribution π according to new trusted information. This possibility distribution π', which induces the new belief set B' after receiving information ϕ, is computed from the possibility distribution π with respect to the previous belief set B ($B' = B * \frac{\tau}{\Phi}, \pi' = \pi * \frac{\tau}{\Phi}$) as follows: for all interpretations ω,

$$\pi'(\omega) = \begin{cases} \frac{\pi(\omega)}{\Pi(\{\phi\})} & \text{if} \quad \omega \models \phi \ \textit{and} \ B(\neg\phi) < 1; \\ 1 & \text{if} \quad \omega \models \phi \ \textit{and} \ B(\neg\phi) = 1; \\ \min\{\pi(\omega), (1-\tau)\} & \text{if} \quad \omega \not\models \phi. \end{cases} \tag{3.2}$$

where τ is the trust degree towards a source about an incoming information ϕ.

3.1.1.2 *BC* Axioms and Rules

Belief context axioms include all axioms from classical propositional logic with weight 1 as in [49]. Since a belief is defined as a necessity measure, all the properties of necessity measures are applicable in this context. Hence, the belief modality in our approach is taken to satisfy these properties that can be regarded as axioms. The following axiom is then added to the belief unit:

$$BC : B(\phi) > 0 \to B(\neg\phi) = 0$$

It is a straightforward consequence of the properties of possibility and necessity measures, meaning that if an agent believes ϕ to a certain degree then it cannot believe $\neg\phi$ at all. Other consequences are:

$$B(\phi \wedge \psi) \equiv \min\{B(\phi), B(\psi)\}$$
$$B(\phi \vee \psi) \geq \max\{B(\phi), B(\psi)\}$$

The inference rules are:

— $B(\neg p \vee q) \geq \alpha, B(p) \geq \beta \vdash B(q) \geq \min(\alpha, \beta)$ (modus ponens)
— $\beta \leq \alpha$, $B(p) \geq \alpha \vdash B(p) \geq \beta$ (weight weakening)

where $\vdash$ denotes the syntactic inference of possibilistic logic.

Let us consider an agent a_1 that represents Bob. Bob believes that his road to work is congested and that there exist other alternative routes that he probably did not know and that are not congested. Using the representation of beliefs that we are proposing, Bob's beliefs can be written as follows:

— $B(Road - to - work - congested) = 1$, meaning that Bob's beliefs that the road to work is congested to a degree equals to 1.

— $B(Exist - alternative - route) = 0.9$,

— $B(No - traffic - in - alternative - route) = 0.7$.

3.1.2 Desire Context

Desires represent a BDI agent's motivational state regardless its perception of the environment. Desires may not always be consistent. For example, an agent may desire to be healthy, but also to smoke; the two desires may lead to a contradiction. Furthermore, an agent may have unrealizable desires; that is, desires that conflict with what it believes possible.

3.1.2.1 The *DC* Language and Semantics

In this context, we make a difference between desires and goals. Desires are used to generate a list of coherent goals regardless to the agent's perception of the environment and its beliefs. Inspired from da Costa Pereira and Tettamanzi [13], the language of DC (L_{DC}) is defined as an extension of a classical propositional language. We define a fuzzy operator D^+, which is associated with a satisfaction degree ($D^+(\phi)$ means that the agent positively desires ϕ) in contrast with a negative desire, which reflects what is rejected as unsatisfactory. For sake of simplicity, we will only consider the positive side of desires in this work, and the introduction of negative desires is left as future work.

In this theory, da Costa Pereira and Tettamanzi [12] use possibility measures to express the degree of positive desires. Let $u(\omega)$ be a possibility distribution called also qualitative utility (e.g., $u(\omega) = 1$ means that ω is fully satisfactory). Given a qualitative utility assignment u (formally, a possibility distribution), the degree to which the agent desires $\phi \in L_{DC}$ is given by:

$$D(\phi) = \Delta([\phi]) = \min_{\omega \models \phi}\{u(\omega)\} \tag{3.3}$$

where Δ is a guaranteed possibility measure that, given a possibility distribution π, is defined as follows:

$$\Delta(\Omega) = \min_{\omega \in \Omega}\{\pi(\Omega)\} \tag{3.4}$$

3.1.2.2 *DC* Axioms and Rules

The axioms consist of all properties of possibility measures such as $D(\phi \vee \psi) \equiv \min\{D(\phi), D(\psi)\}$. The basic inference rules, in the propositional case, associated with Δ are:

— $[D(\neg p \wedge q) \geq \alpha], [D(p \wedge r) \geq \beta] \vdash [D(q \wedge r) \geq min(\alpha, \beta)]$ (resolution rule)
— if p entails q classically, $[D(p) \geq \alpha] \vdash [D(q) \geq \alpha]$ (formula weakening)
— for $\beta \leq \alpha$, $[D(p) \geq \alpha] \vdash [D(p) \geq \beta]$ (weight weakening)
— $[D(p) \geq \alpha]; [D(p) \geq \beta] \vdash [D(p) \geq max(\alpha, \beta)]$ (weight fusion).

Let us consider again our agent a_1 representing Bob. Now suppose that Bob desires to go to work. He would like to take an alternative route without traffic. Besides Bob prefers a route without stops. Such desires can be expressed as follows:
— $D^+(Take-alternative-route) = 0.8$, meaning that Bob desires positively to take an alternative road to a degree equal to 0.8,
— $D^+(No - traffic - in - alternative - route) = 0.8$,
— $D^+(No - stops - in - alternative - route) = 0.75$.
Some of Bob's desires are not consistent with its beliefs which motivates the Goal context detailed in the next Section.

3.1.3 Goal Context

Goals are sets of desires that, besides being logically "consistent", are also maximally desirable, i.e., maximally justified. Even though an agent may choose some of its goals among its desires, nonetheless there may be desires that are not necessarily goals. The desires that are also goals represent those states of the world that the agent might be expected to bring about precisely because they reflect what the agent wishes to achieve. In this case, the agent's selection of goals among its desires is constrained by three conditions. First, since goals must be consistent and desires may be inconsistent, only the subsets of consistent desires can be the potential candidates for being promoted to goal-status, and also the selected subsets of consistent desires must be consistent with each other. Second, since desires may be impossible to realize whereas goals must be consistent with beliefs (justified desires), only a set of feasible (and consistent) desires can be potentially transformed into goals. Third, desires that might be potential candidates to be goals should be desired at least to a degree α. Then, only the most desirable, consistent, and possible desires can be elected as goals.

3.1.3.1 The GC Language and Semantics

The language L_{GC} to represent the Goal Context is defined over the propositional language L extended by a fuzzy operator G having the same syntactic restrictions as D^+. $G(\phi)$ means that the agent has goal ϕ. As explained above, goals are a subset of consistent and possible desires. Desires are adopted as goals because they are justified and achievable. A desire is justified because the world is in a particular state that warrants its adoption. For example, one might desire to go for a walk because he believes it is a sunny day and may drop that desire if it starts raining. A desire is achievable if the agent has a plan that allows it to achieve that desire.

3.1.3.2 GC Axioms and Rules

Unlike desires, goals should be consistent, meaning that they can be expressed by the D_G axiom (D from the KD45 axioms [7]) as follows:

$$D_G \quad GC : G(\phi) > 0 \rightarrow G(\neg\phi) = 0$$

Furthermore, since goals are a set of desires, we use the same axioms and deduction rules as in DC. Goals-beliefs and goals-desires consistency are expressed through bridge rules as we will discuss later on in the thesis.

Considering agent Bob's beliefs and desires, some of its desires cannot become goals. For example, desiring a route without stops cannot be a goal because Bob believes that such a route is not possible. In this scenario, Bob most preferred and possible desire is to take an alternative route without traffic jam.

3.1.4 Social Context

One of the benefits of the BDI model is to consider the mental attitude of the agent in the decision-making process, which makes it more realistic than a purely logical model. However, this architecture overlooks an important factor that influences this attitude, namely the *society* in which an agent lives and acts. There are different ways in which agents can influence each other mental states, e.g., by authority when an agent is influenced by another to adopt a mental attitude whenever the latter has the power to guide the behavior of the former, by trust when an agent is influenced by another to adopt a mental attitude merely on the strength of its confidence in the latter, or by persuasion when an agent is influenced to adopt another agent mental state via a process of argumentation or negotiation. In this work, we will only consider trust as a way by which agents can influence each other.

3.1.4.1 The SC Language and Semantics

In our model, we consider a Multi-Agent system MAS consisting of a set of N agents $MAS = \{a_1, .., a_i, ..a_N\}$. The idea is that these agents are connected in a social network such as agents with the same goal. Each agent has links to a number of other agents (neighbors) that change over time. Between neighbors, we assume a trust relationship holds. The trustworthiness of an agent a_i towards an agent a_j about an information ϕ is interpreted as a necessity measure $\tau \in [0, 1]$, as in Paglieri *et al.* [50], and it is expressed by the following equation:

$$t_{a_i,a_j}(\phi) = \tau \tag{3.5}$$

where $a_i, a_j \in MAS$. Trust is transitive in our model, which means that, it does not hold only between agents having a direct link to each other, but indirect links are also considered. Namely, if agent a_i trusts agent a_k to a degree τ_1, and a_k trusts agent a_j to a degree τ_2, then a_i can infer its trust for agent a_j, and $t_{a_i,a_j}(\phi) = min\{\tau_1, \tau_2\}$.

In large agent networks, agents are often faced to inconsistency and ignorance problems. That is why, it is important to consider trust when designing a MAS in order to control interactions among agents and protect *good* agents from malicious entities. However, apart from trust, in a large group of users, each one equipped with its own intentions, tastes, and opinions, it is natural that also distrust emerges. For this reason, we integrate a distrust value to our model based on Victor *et al.* [51]. A trust network is then defined as a pair (A, R) in which A is the set of users (agents), and R is trust relation such that $A \times A \rightarrow [0,1]^2$ associating to each couple (x, y) of users in A a trust score $R(x, y) = (t, d) \in [0, 1]$ in which t is called the trust degree and d the distrust degree.

3.1.4.2 The SC Axioms and Rules

As the social attitude of the agents is expressed as a trust measure, which is interpreted as a necessity measure, SC axioms include properties of necessity measures as in BC (e.g., $N(\phi \wedge \psi) \equiv \min\{N(\phi), N(\psi)\}$). Concerning distrust, we consider that if an agent is distrusted to a certain degree by another agent towards an information then it cannot be trusted at all and viceversa. For this reason, we add the following axioms:

— $t_{a_i,a_j}(\phi) > 0 \rightarrow d_{a_i,a_j}(\phi) = 0$,

— $d_{a_i,a_j}(\phi) > 0 \rightarrow t_{a_i,a_j}(\phi) = 0$,

When an agent is socially influenced to change its mental attitude, by adopting a set of beliefs and/or desires, the latter should maintain a degree of consistency. Those rules are expressed with bridge rules that link the Social context to the Belief and the Desire contexts.

Let us consider again our agent Bob. Bob is not isolated and may interact with people in its environment, and these agents may influence it (e.g., people using the same road). We introduce, hence, other agents a_3, a_4 and a_5 that represent respectively Alice, Oscar and Mallory. They can (intentionally or not) influence Bob's decision making process, especially when they are considered as trustworthy by Bob. Mallory, for example, may try to maliciously influence the system towards a solution that is not the best for Bob, or it may try to provide information that is not updated in order to have less traffic in its route, e.g., $B(Not - No - traffic - in - alternative - route) = 1$. If Bob accepts this belief, it leads to a change on its goals and hence its plan. In this case, it is interesting to see if the agent network as a whole, using the trust model we are proposing, can avoid this kind of situation, i.e., avoiding the interaction with Mallory and consider it as unreliable. It is also interesting to see if the agent network's welfare increases with the social interaction. All these issues will be the subject of our empirical evaluation.

3.1.5 The communication Context

The communication context is the agent mean to communicate with the external world and other agents. It communicate also information from other contexts, e.g, from the intention context to the the belief one through Bridge rules (*Bridge rule (8)* in Section 3.1.7). Similar to the belief context, the CC uses propositional language with additional connectives. Information added to this context is considered as beliefs.

3.1.6 Planning and Intentions Contexts

The aim of these functional contexts is to extend the BDI architecture to represent plans available to agents and provide a way to reason over them. In this context, we are inspired by Batet and colleagues [1] to represent and reason about plans and intentions. Plans are described using ontologies. Gruber [52] defines an ontology as 'the specification of conceptualizations used to help programs and humans to share knowledge'. According to the World Wide Web Consortium [1] (W3C), ontologies define the concepts and relationships used to describe and represent an area of concern. We use the 5W [2] (Who, What, Where, When, Why) vocabulary which is relevant for describing different concepts and constraints in our scenario. The main concepts and relationships of this ontology are illustrated in Figure 3.3. Using the 5W ontology an intention such as " Running 20 min every 2 days, during 3 months " can be presented as follows:

1. http://www.w3.org/standards/semanticweb/ontology
2. http://ns.inria.fr/huto/5w/

$$
\begin{aligned}
&5w:Process \quad 5w:hasActivity \quad 5w:Running\\
&5w:Running \quad 5w:hasDate[a \quad huto:TemporalExp;\\
&\qquad huto:hasBegin[a \ \ huto:Today];\\
&\qquad huto:hasDuration[a \ \ huto:Duration;\\
&\qquad\qquad huto:hasMonth[a \ \ huto:Month;\\
&\qquad\qquad\qquad huto:number3]];\\
&\qquad huto:exp[a \ \ huto:Cycle;\\
&\qquad\quad huto:every[a\, huto:Day;\\
&\qquad\qquad huto:sample \quad 2]\\
&\qquad\qquad huto:Duration;\\
&\qquad\qquad huto:hasHour[a \ \ huto:Hour;\\
&\qquad\qquad\qquad huto:number2]]]
\end{aligned}
$$

This ontological representation allows besides representing temporal relations (e.g. duration, cycle or repetition) to share and reuse information. Complex requests with temporal and spatial details can be then performed.

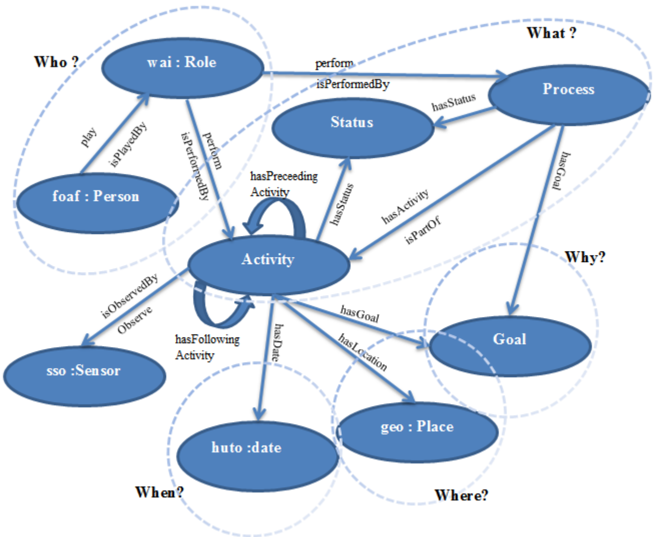


Figure 3.3: The main concepts and relationships of the 5W ontology.

The main task of these contexts is to select plans that maximally satisfy the agent's goals. To go from the abstract notions of desire and belief to the more concrete concepts of goal and plan, as illustrated in Figure 3.4, the following steps are considered: *(1)* new information arrives and updates beliefs or/and desires which trigger goals update; *(2)* these goal changes invoke the Planning Context, whose selection process is expressed by Algorithm 1 (roughly, it looks in the Planning Context for all plans that maximally satisfy these goals);[3] CB and/or CF techniques can be used in the selection process but

3. It is worth noticing that the algorithm complexity is significantly reduced since we discard from the beginning goals without plans.

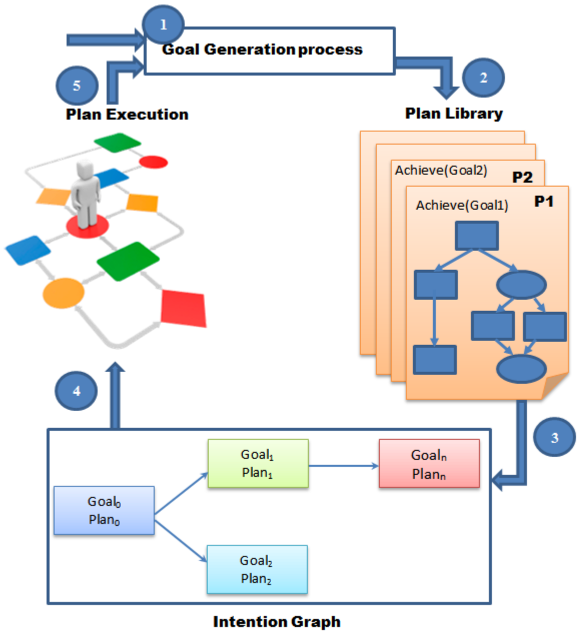


FIGURE 3.4: Planning and Intention Contexts

we leave this issue for further work; *(3)* one or more of these plans is then chosen and moved to the intention structure; and *(4)*, a task (intention) is selected for execution and once executed (successfully or not) this leads to the update of the agent's beliefs *(5)*.

3.1.7 Bridge Rules

There are a number of relationships between contexts that are captured by so-called bridge rules. A bridge rule is of the form:

$$u1 : \phi, u2 : \psi \rightarrow u3 : \theta$$

and it can be read as: if the formula ϕ can be deduced in context $u1$, and ψ in $u2$, then the formula θ has to be added to the theory of context $u3$. A bridge rule allows to relate formula in one context to those in another one.In this section, we present the most relevant rules, illustrated by numbers in Figure 3.2. For all the agents in the MAS, the first rule relating goals to beliefs can be expressed as follows:

$$(1) \vDash GC : G(a_i, \phi) > 0 \rightarrow BC : B(a_i, \neg\phi) = 0$$

which means that if agent a_i adopts a goal ϕ with a satisfaction degree equal to β_ϕ then ϕ is believed possible to a degree β_ϕ by a_i. Concerning rule (2) relating the goal context

Data: G* // $G^* = \{\phi_1, \phi_2, .., \phi_n\}$, a list of elected goals
Result: S //S is a list of plans
$m \leftarrow 0; S' \leftarrow \emptyset; G' \leftarrow \emptyset;$
for ***each*** ϕ_i *in* G^* **do**
    //Search in the PC for a plan satisfying ϕ_i
    $S_{\phi_i} \leftarrow SearchInPC(\phi_i);$
    if $S_{\phi_i} \neq \emptyset$ **then**
        //Discard goals without plans
        $Append(G', S_{\phi_i});$
    end
end
for i *in* $1..Lenght(G')$ **do**
    //Combination of i elements in G'
    $S' \leftarrow Combination(G', i);$
    for j *in* $1..Length(S')$ **do**
        if $S'[j] \neq \emptyset$ **then**
            //Compute the satisfaction degree of S' *using the Goal logic operator*
            $\alpha_i = G(S'[j]);$
            //Select the maximum α_i
            if $\alpha_i > m$ **then**
                $m \leftarrow \alpha_i;$
                Initialize(S);
                Append(S, S');
            else
                if $\alpha_i = m$ **then**
                    Append(S,S');
                end
            end
        end
    end
end
Return S;

Algorithm 1: *RequestForPlan* Function

to the desire context, if ϕ is adopted as goal then it is positively desired with the same satisfaction degree.

$$(2) \vDash GC : G(a_i, \phi) = \delta_\phi \rightarrow DC : D^+(a_i, \phi) = \delta_\phi$$

An agent may be influenced to adopt new beliefs or desires. Beliefs coming from other agents are not necessarily consistent with the agent's individual beliefs. This can be expressed by the following rule:

$$(3) \vDash BC : B(a_j, \phi) = \beta_\phi, SC : T_{a_i,a_j}(\phi) = t \rightarrow BC : B(a_i, \phi) = \beta'_\phi$$

where β'_ϕ is calculated using Equation 3.2 with $\tau = min\{\beta_\phi, t\}$ to compute the possibility distribution, and Equation 3.1 to deduce the Belief degree.

Data: B,D
Result: G^*, γ^*
$\bar{\gamma} \leftarrow 0$;
repeat
  Compute $G_{\bar{\gamma}}$ by Algorithm 3;
  if $G_{\bar{\gamma}} = \emptyset$ **then**
    //Move to the next more believed value in B
    $$\bar{\gamma} \leftarrow \begin{cases} \min\{\alpha \in Img(B) \mid \alpha > \bar{\gamma}\} & \\ 1 & if \nexists \alpha > \bar{\gamma} \end{cases}$$
  end
until $\bar{\gamma} < 1$ *and* $G_{\bar{\gamma}} = \emptyset$;
$\gamma^* = 1 - \bar{\gamma}, G^* = G_{\bar{\gamma}}$;

Algorithm 2: The goal election function.

Data: B, D, $\bar{\gamma}$
Result: $G_{\bar{\gamma}}$
//$Img(D)$ is the level set of D, i.e., the set of membership degrees of D
$\delta \leftarrow \max Img(D)$;
//Find the most desired δ-cut D_δ of D which is believed possible
while $\min_{\psi \in D_\delta} B(\neg\psi) \leq \bar{\gamma}$ *and* $\delta > 0$ **do**
  //while not found, move to the next lower level of desire
  $$\delta \leftarrow \begin{cases} \max\{\alpha \in Img(D) \mid \alpha < \delta\} & \\ 0 & if \nexists \alpha < \delta \end{cases}$$
end
if $\delta > 0$ **then** $G_{\bar{\gamma}} = D_\delta$;
else $G_{\bar{\gamma}} = \emptyset$;

Algorithm 3: Computation of $G_{\bar{\gamma}}$.

Similarly to beliefs, desires coming from other agents need not to be consistent with the agent's individual desires. For example, an agent may be influenced by another agent to adopt the desire to smoke, and at the same time having the desire to be healthy, as shown by the following rule:

$$(4) \models DC : D^+(a_j, \psi) = \delta_\psi, SC : T_{a_i,a_j}(\psi) = \tau \rightarrow DC : D^+(a_i, \psi) = \delta'_\psi$$

where $\delta'_\psi = min\{\delta_\psi, \tau\}$.

Desire-generation rules can be expressed by the following rule:

$$(5) \models BC : min\{B(\phi_1) \wedge ... \wedge B(\phi_n)\} = \beta, DC : min\{D^+(\psi_1) \wedge ... \wedge D^+(\psi_n))\} = \delta \rightarrow DC : D^+(\Psi) \geq min\{\beta, \delta\}$$

Namely, if an agent has the beliefs $B(\phi_1) \wedge ... \wedge B(\phi_n)$ with a degree β, and it positively desires $D^+(\psi_1) \wedge ... \wedge D^+(\psi_n)$ to a degree δ, then it positively desires Ψ to a degree greater or equal to $min\{\beta, \delta\}$. According to [13], goals are a set of desires that, besides being logically 'consistent", are also maximally desirable, i.e., maximally justified and possible. This is expressed by the following bridge rule:

$$(6) \vDash BC : B(a_i, \phi) = \beta_\phi, DC : D^+(a_i, \psi) = \delta_\psi \rightarrow GC : G(\chi(\phi, \psi)) = \delta$$

where $\chi(\phi, \psi) = \mathit{ElectGoal}(\phi, \psi)$, as specified in Algorithm 2, is a function that allows to elect the most desirable and possible desires as goals. If *ElectGoal* returns $\emptyset$, then $G(\emptyset) = 0$, i.e., no goal is elected.

As expressed by the bridge rule above, once goals are generated, our agent will look for plans satisfying goal ϕ by applying the *RequestForPlan* function, and do the first action of the recommended plan.

$$(7) \vDash GC : G(a_i, \phi) = \delta, PC : RequestForPlan(\phi) \rightarrow IC : I(act_i, PostConditon(act_i))$$

where *RequestForPlan* is a function that looks for plans satisfying goal ϕ in the plan library, as specified in Algorithm 1. Rule (8) means that if an agent has the intention of doing an action act_i with $\mathit{PostCondition}(act_i)$ then it passes this information to the communication unit and via it to other agents and to the user.

$$(8) \vDash IC : I(act_i, PostConditon(act_i)) \rightarrow CC : C(does(act_i, PostConditon(act_i)))$$

If the communication unit obtains the information that some action has been completed then the agent adds it to its beliefs set using rule (3) with $B(PostConditon(act_i)) = 1$.

To show the applicability of our Multi-Agent BDI framework, an experimental evaluation is proposed using the NetLogo Platform, as detailed in the next Section.

3.1.8 Experiment

In agent-based systems with spatial reasoning and social behavior, a visual output is needed to display the agents' interactions in two or three dimensional spaces. The Netlogo graphical user interface offers the possibility to design agents with different shapes and positions. Each agent in the simulation environment is a multi-context BDI

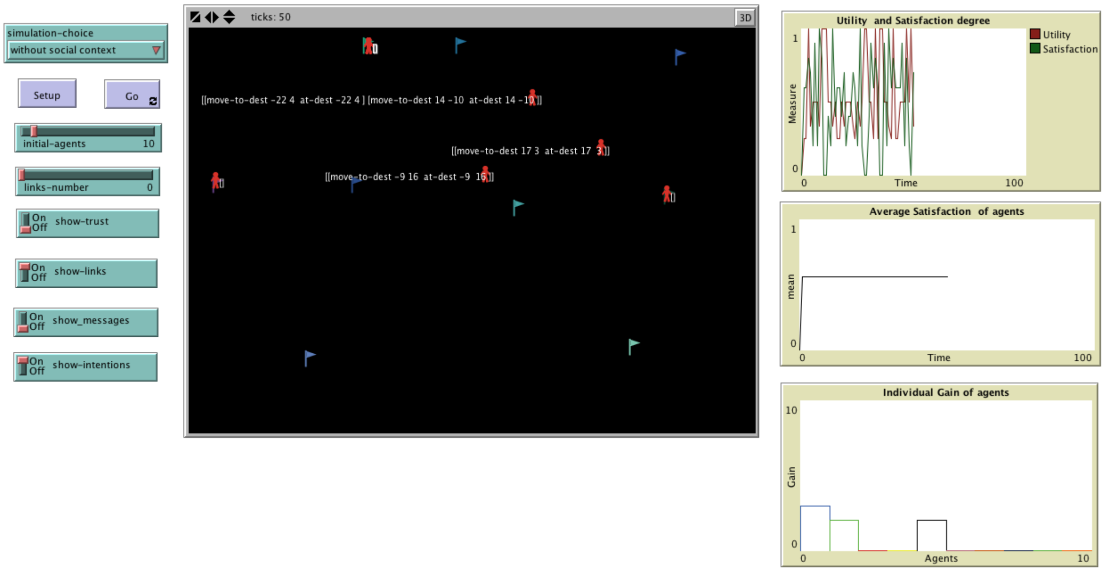


FIGURE 3.5: The user interface of our Multi-Agent simulation environment in Netlogo. The person icon represents an agent in the MAS. Flags represent destinations in which agents can go. Labels represent agent intentions which consist of two elements: the name, mapped to a NetLogo command, and a done-condition, mapped to a NetLogo reporter. Intentions are stored in a stack, and are popped out when to be executed. If the done-condition is satisfied, the intention is removed and the next intention is popped out consecutively. The Figure shows also, on the right-hand side, how the graphs are updated dynamically as the program runs.

agent implementing the behavior formally detailed in the previous Sections. An agent represents a user with different desires and beliefs that are randomly initialized. The aim of the simulation is to compute a recommendation based on a user initial set of beliefs and desires, and to see how our agent will adapt the recommendation, with a particular interest in the following two cases:

— the agent is part of a social network (*social agent*), i.e., it has relationships with other agents similar to it,

— the agent is considered as a *solitary agent*, i.e., it has no interaction with the other agents.

Plans consists of a list of activities representing the fact of moving from one destination to another. Each destination contains some rewards that the agent obtains if it reaches that destination. The amount of rewards for each agent is random. Once rewards are gained, an agent will broadcast information about the number of remaining rewards in the correspondent destination to similar agents. These agents will decide to accept or not this recommendation according to the trust degree in the sender, and whether there is any information in their knowledge base (desire or belief base) that contradicts this one. If an agent decides to accept the recommendation, then it adds this information to its desire base, and then it triggers the recalculation of its intentions according to the updated desire base. The degree to which the agent believes or desires this recommendation is updated according to the degree of trust towards the sender of the proposal and to the its degree of belief or desire.

3.1.9 Experimental Setup

Table 3.1 summarizes the parameters that can be varied for different use cases. As shown in Figure 3.5, agents are initially randomly distributed in the space (patches in NetLogo). They also have different profiles (desires, trust degrees, positions, ...). Links are also created randomly between agents according to an initial link number set at the beginning of the simulation through the user interface (visualized on the left-hand side of Figure 3.5).

We used Netlogo v. 5.3.1 to implement our simulation. For the BDI behavior and the communication context, we used two available NetLogo libraries [53], one for BDI-like agents and the other for ACL-like communication, allowing the development of goal-oriented agents, communicating with FIPA-ACL messages. We implemented the rest of the behavior of the agents using the NetLogo language with some extensions.

Table 3.1: The scale and distribution of parameters in the simulation.

Parameters	Scale	Distribution
Number-of-agents	0-100	Random
Desires	0-50	Random
Beliefs	0-100	Random
Intentions	0-10	Random
Links	0-100	Random
Gain	0-50	Random

The objective here is to assess the effects of these agents on the system as a whole (and not only to assess the effect of individual agents on the system).

3.1.10 Results and discussion

The model and experimental data were analyzed using the RNetLogo extension [54]. Once the experiment is set up, each agent has a list of random desires, whilst beliefs are empty at the beginning. According to these desires and the aforementioned behavior, an agent calculates the recommendation which has a plan as output. This plan becomes the agent's intention, and the agent will execute it. In the case of a solitary agent, it executes its plan without any change. Only a new belief from an external source that does not contradict the agent initial belief can make it change its intention. In the other case, i.e., a social agent, similar agents will communicate a set of proposed recommendations with the aim to influence the others to change their beliefs or desires. If the recommendation is accepted, the agent recalculates its intentions based on the received recommendation then following a new plan. Metrics such as utility or satisfaction are calculated using the following equations:

$$\text{utility}(p) = \frac{\sum_{i \in G_S} g(i)}{\sum_{j \in D_{\text{initial}}} d(j)}$$

where G_S is a set of goals satisfied by a plan p, and D_{initial} is a set of initial desires of an agent.

$$\text{satisfaction-degree}(p) = \max\{G(\phi_i), i \in [0, n]\}$$

where n is the number of goals satisfied by a plan p. The utility measure estimates how much the user needs (desires) to match the recommendation (plan). The satisfaction degree, as its name suggests, computes the user satisfaction about a recommendation based on its initial degrees of desires.

The mean gain of the agents is also reported, and results are showed in Figure 3.7. We can see that agents within a social context, i.e., agents that communicate in order to influence each other, accrue more gain most of the time in comparison with those without a social context. These results demonstrate that a social population could have a greater social welfare than a non-social one, when agents have similar interests.

Concerning social agents, as time passes, a number of links among agents are created based on similarities between them. Geo-localization is implemented in our experimental setting in a similar way. That is, if two agents are in the same location at a specific time instant, a link is created between them. An agent can then exchange with its neighbors its desires or beliefs. The resulting network is captured in Figure 3.6 showing the agent network evolution over time. We can see that links increase over time, and we reach a fully connected network at time 100. This means that all agents in the network can exchange their desires and beliefs with each other. The acceptance of such a proposal depends on the agent knowledge base (i.e., its desires and beliefs) and the trust degree of the sender agent. Now that we have such networks, it is interesting to verify whether agents in "communities" are more likely to have better performance than the others.

For comparison, we calculate the average satisfaction degree and utility over time for 50 agents in the case of solitary and social agents. One may expect that the probability of gaining utility will decrease with exchanging messages. Figure 3.8 confirms this

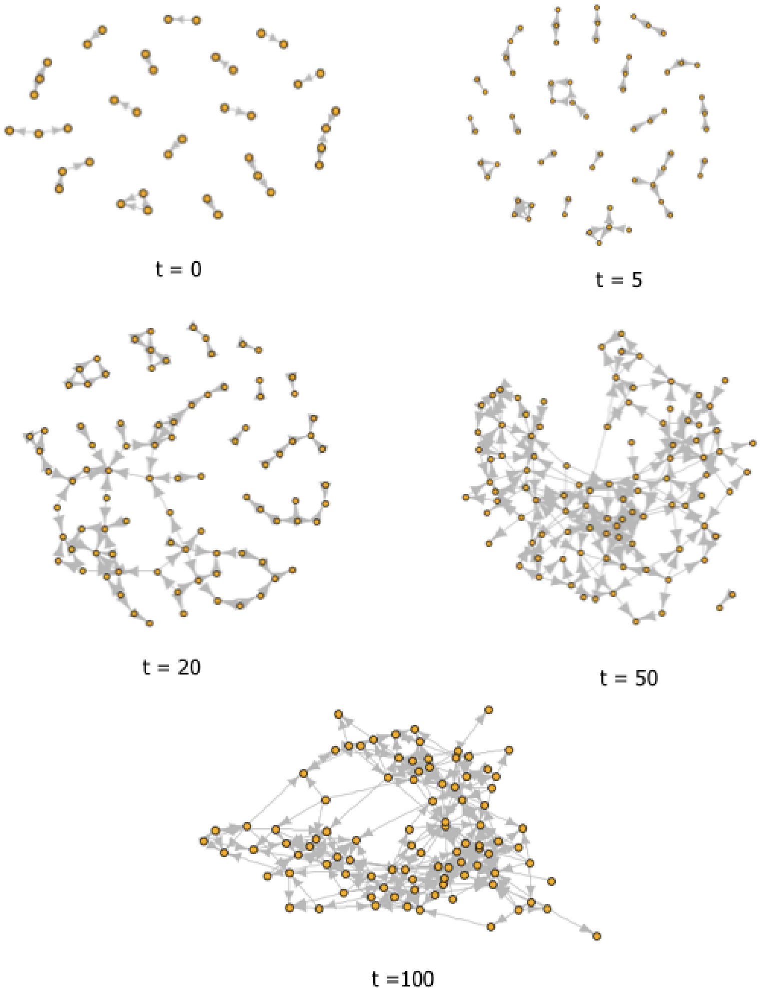


FIGURE 3.6: The Multi-Agent network at different time-points (start with 100 agents and 22 edges). Time here is equivalent to NetLogo ticks. Nodes represent agents and edges represent similarity between them. Graphs are generated using igraph for R (http://igraph.org/r/).

expectation. It shows that utility augments considerably within social agents compared to the utility within solitary ones. We notice that the average utility is the same over time for solitary agents. We can deduce that exchanging beliefs and desires increases, on average, the agents' utility.

In Figure 3.9, we can see the average satisfaction of agents about recommendations (plans). This average is higher within social agents than within individual ones. We can conclude that agents get more satisfaction collectively from exchanging information.

These results provide for agents further motivation to engage in communications with similar trustworthy agents and support our modeling choices. It is also interesting to note how communities of agents (e.g., agents with similar interests) likely to be influenced are more efficient collectively than solitary agents.

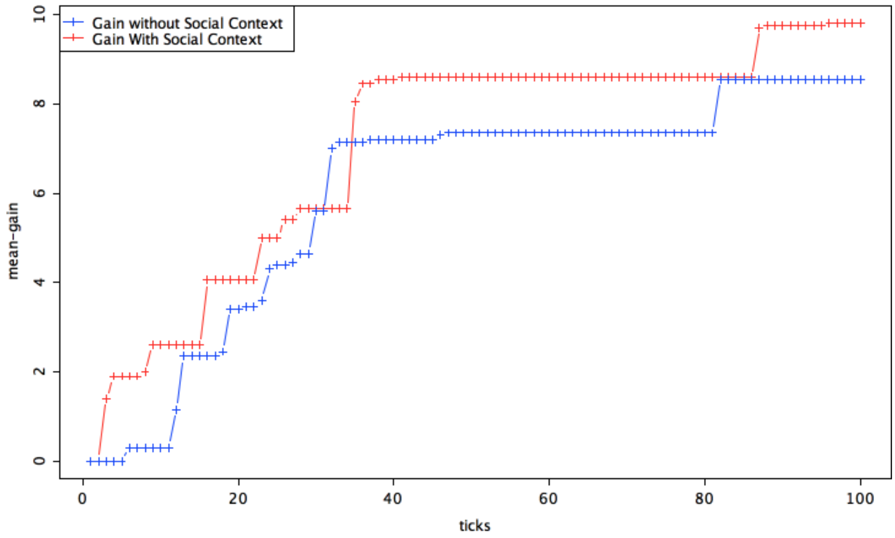


FIGURE 3.7: Mean gain of agents with and without a social context.

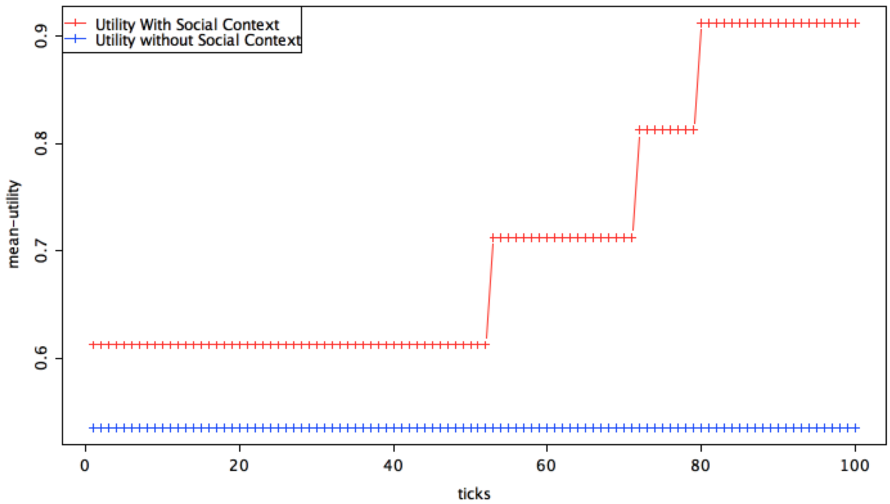


FIGURE 3.8: Mean utility of agents with and without social context.

In addition, we are also interested in studying how the system behaves in case agents communicate incorrect information, i.e., how fast and how far will these messages propagate? Can the authenticity of the message be detected with this agent's behavior and, if it is the case, how is the trust distribution affected? In order to experimentally evaluate whether incorporating the trust/distrust model can indeed enhance the performance of the model, we run the simulation scenario with two different settings:

1. agents communicate with each other without considering trust knowledge about other agents.
2. similar agents initially trust each other to a certain random value. Consequently, none of the agents distrust other agents at the beginning of the simulation. The value of trust and distrust can change over time. The trust value increases with an

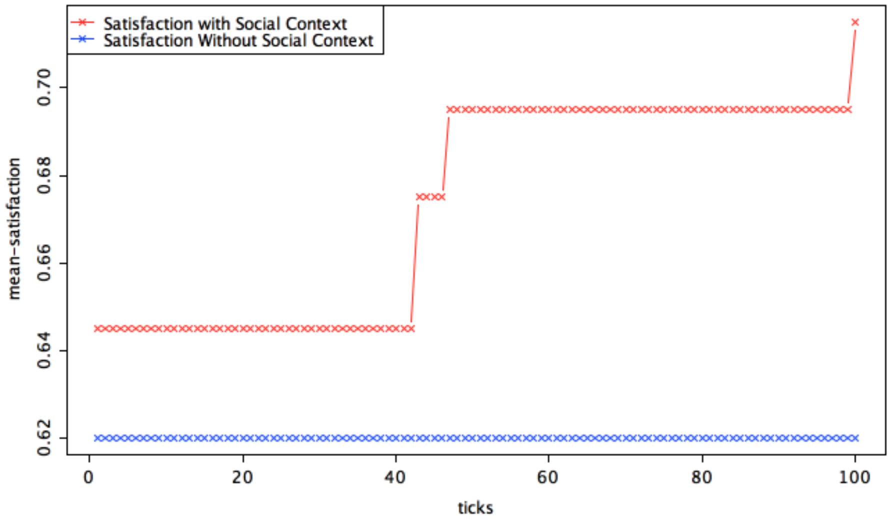


FIGURE 3.9: Mean satisfaction degree of agents with and without social context.

α coefficient if the agent provides some reliable information. In contrast, distrust is set to 1 if the agent provides erroneous information.

We also studied the propagation of the error in the agents system in these two cases. In both cases, we use the same parameters specified in Table 3.1. At $t = 50$ ticks, we ask an agent (choosed randomly) to send a false information to all its connected agents, i.e., similar agents. This information is sent over time from different random agents.

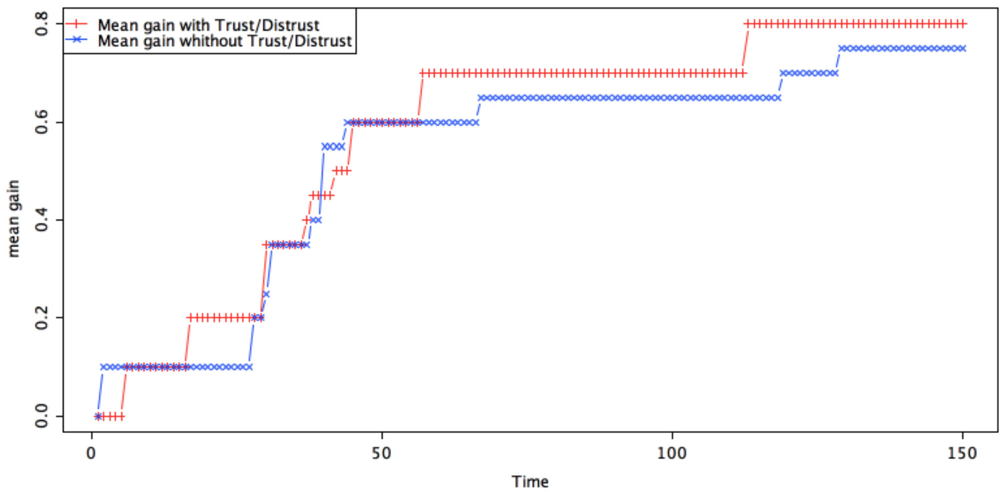


FIGURE 3.10: Average Gain in the agent network with and without trust/distrust

In order to capture the impact of the use of the trust/distrust information in the MAS, we report the average gain of the whole system in both cases. Results are showed in Figure 3.10. We notice that this average remains almost the same before sending the erroneous information. Once we start propagating the error, this average arises

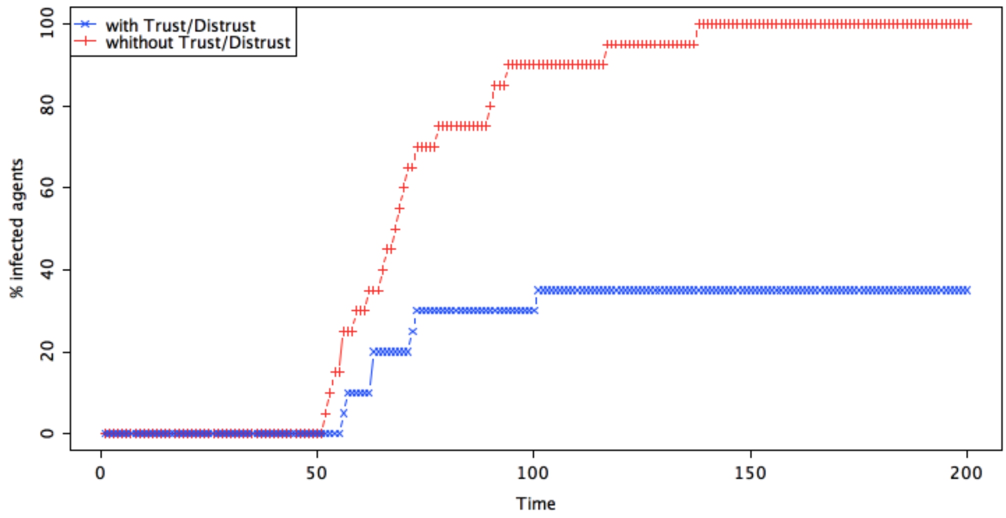


FIGURE 3.11: Error propagation in the agent network with and without trust/distrust couple

when using the trust/distrust information. Integrating trust/distrust to the agent model enhances then the agents gain.

To demonstrate that the use of trust/distrust limits the propagation of error in the MAS, we also performed a second set of experiments, following a similar procedure as for the first experimetal setting. Figure 3.11 reports about the percentage of agents that received the false information, added it to their desire base and probably to their intentions, which are called *infected agents*. Once we start propagating the error, the impact in the agent network is immediate in both cases: the number of infected agents increases but in a different way. In fact, the error propagates faster in the model without trust/distrust, and all agents in the network finish by receiving the false information and adding it to their desire base. In the other case, this number remains acceptable compared to the trust-less counterpart, and finishes by being normalized to 35% at $t = 100$. This tells us that the use of the trust/distrust model limits the error propagation and consequently allows agents to achieve their goals in less time with more gain.

These results confirms our starting hypothesis that involving trust/distrust in the recommendation process enhances the social welfare of the MAS and the quality of the recommendation.

3.1.11 The Traffic Scenario

We implemented Bob's real world example detailed throughout Section 3.1 in the Netlogo environment. The agents in the MAS aim at reaching a work destination using the less congested route. In order to see the impact of Mallory as a malicious and selfish agent trying to alleviate the traffic in its route by sending an erroneous information, we create a cycle consisting of a home-work and work-home routes. First, as showed in Figure 3.12, Bob and the other agents take the information sent by Mallory into consideration and change their route. Unfortunately, they were all misled by Mallory who succeeded to alleviate the traffic in its route. With the trust/distrust model, agents discover that Mallory is not reliable and then consider it as distrusted.[4]

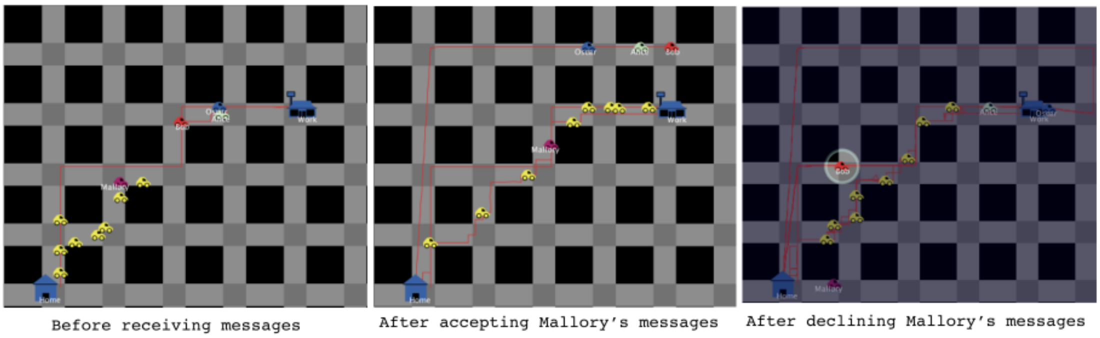


FIGURE 3.12: The traffic scenario simulation in the Netlogo environment at different time points: at the beginning of the simulation (top left), Bob takes an alternative route proposed by the system. After receiving Mallory's messages (in the middle), Bob decides to change its route and takes another longer route. After a while Bob figures out that it was not the best choice and consider Mallory's agent as unreliable (top right).

For simplicity reasons, we argue for an update of intentions once the agent has an empty intentions stack. In the other case, i.e., updating intentions at random time in a random place, the Dijkstra algorithm can be considered to compute the optimal route to work based on `http://modelingcommons.org/browse/one_model/4485`.

Many real-world scenarios such as the traffic scenario presented above require additional features for representing and reasoning about spatial and temporal knowledge considering also their vague connotation. To enable our agent model to represent and reason about these features we propose in the next section an extention of **CARS** agent framework with fuzzy spatio-temporal representation.

4. The demonstration is available online in this link: `http://modelingcommons.org/browse/one_model/4752`.

3.2 An uncertain Spatio-temporal Cognitive Agent-based Recommender Framework

3.2.1 Fuzzy sets for representing imprecise spatio-temporal beliefs and desires

Spatio-temporal data are often affected by imprecision and uncertainty [55] due to several reasons. Spatial uncertainty refers to positional accuracy (e.g., location of an individual or a car). Temporal uncertainty states whether temporal information describes well a spatial phenomena. A fuzzy set, because of its ability to represent degrees of membership, is more suitable for modeling geographical entities. In a GIS database, real world objects can be represented by the degrees of membership to multiple classes or objects.

Representing only spatial or temporal dimension is not sufficient to model and analyse such phenomena. Modeling change involves incorporating both dimensions simultaneously. In this work, we adopt a dual representation of dynamic spatial information proposed by Bordogna *et al.* [56]. In this approach, they introduced two representations:

— the first one in which we have a precise spatial reference and indeterminate or vague time reference, e.g. if i leave home now, i should be at work around 8 pm,

— and the second one defined with a precise time reference and a fuzzy spatial one, e.g. An accident has just occurred in between Route A and Route B.

According to [56], a spatial dynamic object can be represented in the first case as a set of pairs (τ_i, o_i):

$$o_d := \{(\tau_1, o_1), .., (\tau_i, o_i), ..., (\tau_n, o_n)\}$$

where τ_i is the time fuzzy validity range associated with the spatial object o_i. The semantics of τ_i is defined by a triangular membership function centred in t_i (see Figure 3.13). In the same way, a spatial object with precise time reference is defined by a set of pairs (t_i, σ_i), where σ_i stands for the spatial validity of the observed phenomenon at time instant t_i represented as a triangular membership function.

In order to reason about such information, we need a mechanism to represent also qualitative relationships between spatio-temporal entities. For this reason, we propose a fuzzy RCC-8 and an extension to Allen's intervals to support fuzziness.

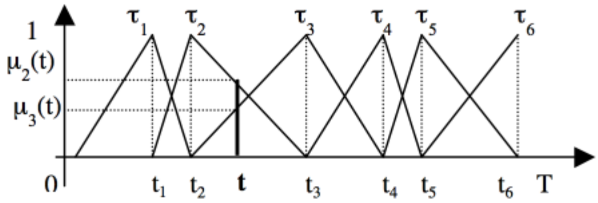


FIGURE 3.13: Fuzzy time membership function.

3.2.1.1 Fuzzy Allen's Intervals

The twelve relations defined by Allen for simple time intervals presented in Section 2.5 are generalized for modeling fuzzy time relations. Each basic relation can be defined in terms of endpoint relations defined in Table 2.2. Using the extension principle, a fuzzy temporal relation is defined. For example, the fuzzy relation d_f is introduced for the simple temporal relation d (during), as follows:

$$X d_f Y \Leftrightarrow (X^- >_f Y^-) \wedge (X^+ <_f Y^+)$$

and the corresponding degree of confidence, using the extension principle, can be expressed as:

$$\mu_{X d_f Y} = \min(\mu_{X^- >_f Y^-}, \mu_{X^+ <_f Y^+})$$

All the values X and Y can be generalized to fuzzy values and represented by fuzzy triangular numbers. Based on the extension principle, we define first the confidence degrees of the fuzzy relations $\geq_f$ and $\leq_f$, in order to deduce respectively the one of $>_f$, $<_f$ and $=_f$. Suppose we have two fuzzy intervals A and B defined by triangular fuzzy functions as follows: $A = (a_1, a_2, a_3)$ and $B = (b_1, b_2, b_3)$. By applying the extension principle, we can deduce the following fuzzy relations:

$$\mu_{A \leq_f B} = \begin{cases} 0 & \text{if} \quad a_1 > b_3 \\ \frac{b_3 - a_1}{b_3 - a_1 + a_2 - b_2} & \text{if} \quad a_1 \leq b_3, b_2 < a_2 \\ 1 & \text{if} \quad a_2 \leq b_2 \end{cases} \tag{3.6}$$

$$\mu_{A \geq_f B} = \begin{cases} 0 & \text{if} \quad b_1 > a_3 \\ \frac{a_3 - b_1}{a_3 - b_1 + b_2 - a_2} & \text{if} \quad b_1 \leq a_3, b_2 > a_2 \\ 1 & \text{if} \quad b_2 \leq a_2 \end{cases} \tag{3.7}$$

From Equations 3.6 and 3.7 we can deduce the confidence degree of relations $>_f$, $<_f$ and $=_f$ as follows:

$$
\begin{aligned}
A <_f B &= A \leq_f B \wedge \neg(A =_f B) \\
A >_f B &= A \geq_f B \wedge \neg(A =_f B) \\
(A =_f B) &= A \leq_f B \wedge A \geq_f B
\end{aligned}
$$

Example 3.1. *Let us consider* $A = (8, 9, 10)$ *and* $B = (8.5, 9.5, 10.5)$ *representing two fuzzy time-points. We can compute the degree of confidence of this fuzzy temporal relation "A occurs at approximately the same time as B" using Equation 3.6 and Equation 3.7 as follows :* $\mu_{A=_f B} = \mu_{A\leq_f B \wedge A\geq_f B} = \min(\mu_{A\leq_f B}, \mu_{A\geq_f B}) = \min(1, 0.75) = 0.75$.

3.2.1.2 Fuzzy Topological Relations

The eight binary topological predicates for simple regions (Section 2.6) are generalized for modeling fuzzy topological relations. Based on the approach proposed by Schockaert *et al.* [3] and the definition of the RCC relations in Table 2.2, we present here an approach for modelling imprecise spatial information when regions are represented as fuzzy sets.

Let U be a nonempty set (representing regions), and **C** a reflexive and symmetric binary fuzzy relation on it modeling connection. Several other topological relations can be defined based on this relation. These include the RCC8 basic relations DC, EC, PO, EQ, TPP, NTPP, and the converses of TPP and NTPP (see Table 3.2 for their definitions). Note that we adopt, following [57], the Łukasiewicz-norm T_w and its corresponding implicator I_{T_w} to generalize the standard logical conjunction and implication. In addition, we chose this logic for its convenience, especially regarding the implication function. The implicator corresponding to the Łukasiewicz t-norm is defined by: $I_{T_W} = \min(1, 1 - x + y)$. In fact, the minimum operator does not eliminate values arbitrary, leaving thus more uncertainty. For simplicity, we write I_W instead of I_{T_W} in the remainder of the Section.

Using this formalism, we can for example calculate a fuzzy spatial relation "p is precisely located far from q". Knowing the location of p and q, we can calculate their fuzzy position using Equation 3.8. We can then calculate the degree to which those two locations are connected, and consequently, their degree of disconnection: $DC(p, q) = 1 - C(p, q)$.

TABLE 3.2: Fuzzy RCC definitions from [3]

Name	**Definition**	**Fuzzy Definition**
$DC(x,y)$	$\neg C(x,y)$	$1 - C(x,y)$
$P(x,y)$	$\forall z \in U, C(z,x) \to C(z,y)$	$\inf_{z\in U} I_W(C(z,x), C(z,y))$
$PP(x,y)$	$P(x,y) \wedge \neg P(y,x)$	$\min\left(P(x,y), 1 - P(y,x)\right)$
$EQ(x,y)$	$P(x,y) \wedge P(y,x)$	$\min\left((P(x,y), P(y,x)\right)$
$O(x,y)$	$\exists z \in U, P(z,x) \wedge P(z,y)$	$\sup_{z\in C} T_W(P(z,x), P(z,y))$
$DR(x,y)$	$\neg O(x,y)$	$1 - O(x,y)$
$PO(x,y)$	$O(x,y) \wedge \neg P(x,y) \wedge \neg P(y,x)$	$\min\left(O(x,y), 1 - P(x,y), 1 - P(y,x)\right))$
$EC(x,y)$	$C(x,y) \wedge \neg O(x,y)$	$\min\left(C(x,y), 1 - O(x,y)\right)$
$NTP(x,y)$	$\forall z \in U, C(z,x) \to O(z,y)$	$\inf_{t\in U} I_W(C(z,x), O(z,y))$
$TPP(x,y)$	$PP(x,y) \wedge \neg NTP(x,y)$	$\min\left(PP(x,y), 1 - NTP(x,y)\right)$
$NTPP(x,y)$	$PP(x,y) \wedge NTP(x,y)$	$\min(1 - P(x,y), NTP(x,y))$

3.2.2 Fuzzy spatio-temporal belief representation and reasoning

In order to represent an imprecise spatio-temporal belief or desire such as "An accident occurred around 8 PM between road A and road B" or "I want to be at work before 9 AM", we combine the RCC spatial relations with Allen's temporal relations. The degree to which this belief is true is computed using the minimum between the degrees of confidence of the spatial belief and the temporal one, respectively. For representing a spatio-temporal belief, we annotate spatial formula with temporal information, meaning that a spatial formula is true during a time interval or at a specific time point. In other words, it can be written as follows:

$$X\ DC_I\ Y, Y\ PO_J\ Z$$

where X and Y represent two different regions or moving objects, and I and J are time intervals. This formula means that X is disconnected from Y during time interval I, and Y is part of Z during time interval J. The following example shows a concrete example of our combined fuzzy spatio-temporal belief representation in the traffic scenario.

Example 3.2. *Let us consider again the belief "An accident (A) occurred around 8 PM(t_1) between road A(R_A) and road B(R_B)". It can be formalized as follows:*
$(A\ PO_{t_1} R_A) \wedge\ (A\ PO_{t_1} R_B)$ *and its degree of belief is:*

$$\begin{aligned} &B((A\ PO_{t_1} R_A) \wedge\ (A\ PO_{t_1} R_B)) \\ &= \min\{B(A\ PO_{t_1} R_A), B(A\ PO_{t_1} R_B)\} \end{aligned}$$

Later, one can reason about temporal intervals or time-points to infer relevant information such as being at the same time nearby the accident place. This spatio-temporal belief is essential for an agent to decide or not to reconsider its intention in case the degree of confidence of this belief is high. However, this belief is no longer useful after a certain time period, or if the accident is not placed on the agent's route (i.e., intentions).

3.2.3 Experiment

In this section, we present the evaluation of the CARS recommendation system equipped with the fuzzy spatio-temporal belief representation. The purpose of the evaluation is to quantify the gain of agents, in terms of execution and limited waiting time, to reach their

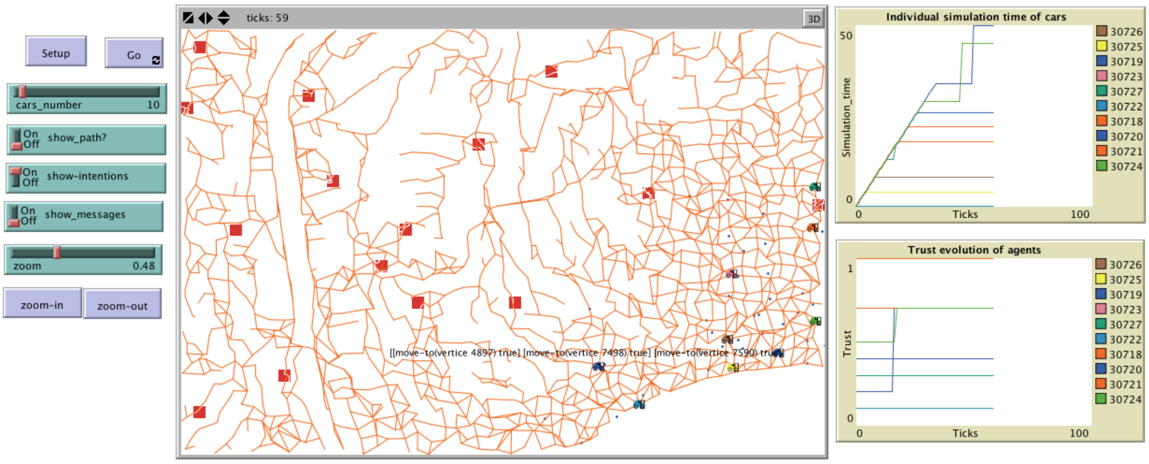


FIGURE 3.14: The user interface of the agent-based simulation in NetLogo. The central part shows the agent's environment constituted of roads. Blue points represent Electric Vehicle charging stations. An agent is represented by a car. Red squares represent accidents. Labels represent an agent intention, which consists of two elements: the name, mapped to a NetLogo command, and a done-condition, mapped to a NetLogo reporter. Intentions are stored in a stack, and are popped out when they are to be executed. If the done-condition is satisfied, the intention is removed and the next intention is popped out consecutively. The figure shows also, on the right-hand side, how the graphs are updated dynamically as the program runs. The left-hand pane shows some setup parameters.

goals, by exchanging spatio-temporal beliefs and desires. To this aim, we propose to test the proposed model in a real-world scenario where spatio-temporal knowledge represents a crucial factor in the user decision making process. In this evaluation, different agent's strategies are considered, following the ideas we proposed in [58]:

— *individual agent strategy*: agents behave individually without taking into account any information coming from other agents. Only information from external resources are considered in this case, e.g., data from the Traffic Message Channel (TMC).

— *social agent strategy*: agents are part of a social network and communicate with the other agents in the network by exchanging their own beliefs and desires. Agents fully trust all other agents in the network.

— *social distrustful agent strategy*: agents are part of a social network, but they consider also the trustworthiness degree of the other agents, when they exchange messages. Agents accept information only from trustworthy agents. An agent is considered as deceitful if the information it provides is repeatedly proven to be false.

3.2.3.1 Scenario

In order to evaluate the applicability of the proposed model in a real-world application, we propose the following scenario. Agent a_1 uses an electric car, and needs to reach an electric public charging point. Like most road users, a_1 usually consults web-based or mobile mapping services before the trip to determine the nearest charging station and to avoid possible traffic jams. Knowing where to get to and estimating the time needed for the journey, a_1 can plan its trip. Thus, it selects a course of actions that will result in reaching its destination before the battery of its car goes out of charge. It chooses a route to follow and a time to leave so that it can arrive by a desired arrival time. Once the trip is planned, it can be executed. As long as a_1 has not found any obstacle within the journey, it can keep executing its original plan. However, it just found that a certain road on its route is closed due to an accident (other city events such as soccer games or music concerts can be considered as well). As a_1 is not able to drive through that road anymore, it has to reconsider its options and find an alternative route to reach its destination while taking into account its battery life (hence its arrival time).

3.2.3.2 Implementation

In agent-based systems with spatial reasoning and social behavior, a visual output is needed to display the agents' movements and interaction in two- or three-dimensional spaces. To implement our scenario, we decided to use NetLogo, as it also provides support for the BDI architecture and the FIPA Agent Communication Language. The spatial module is implemented using the Geographic Information Systems (GIS) extension for Netlogo. [5] We used data about the road network and Electric Vehicle (EV) charging points from the Nice city open geographical database [6] in shapefile format (i.e., the format supported by the GIS Netlogo extension). The resulting environment of agents is shown in Figure 3.14.

In order to adapt a fuzzy topological relation to a GIS vector data model, we assume that crisp regions are a set of trapezoidal shapes containing a finite sequence of line segments. To simplify the representation, we use a Gaussian function distribution as an approximation of the trapezoidal distribution. Then, the membership function $\mu(x, y)$ of a spatial object with coordinates (x, y) is defined by the following equation:

$$\mu_{x,y} = e^{-k_d|(x-x_R)+(y-y_R|^2}, \tag{3.8}$$

5. https://ccl.northwestern.edu/netlogo/docs/gis.html
6. http://opendata.nicecotedazur.org/data/

where x_R and y_R are the coordinates of a landmark point, and k_d corresponds to a flattening coefficient defined according to the user description (d) of a belief. We define then different coefficients for $k_{\text{precisely}}$, $k_{\text{approximately}}$, k_{near}, k_{around}. An example of this distribution run is visualized in Figure 3.15.

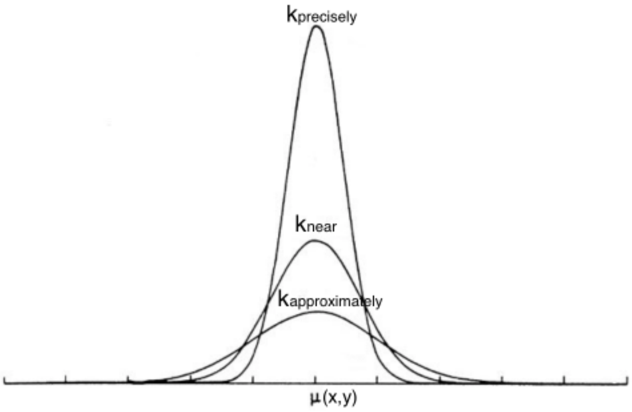


FIGURE 3.15: Example of the Gaussian distribution

Agents in this simulation are spatial entities (moving cars) in an environment (the road network of the Nice city) which may change their location and attributes as time goes by. At the beginning of the simulation, each agent has a desire. As defined in our scenario, the desire of an agent is to go to the nearest EV recharge point. A recommended plan is proposed to the agent following the multi-context approach to the deliberation of agent behavior proposed by Othmane *et al.* [58, 59]. Once the agent starts executing this plan, we trigger at different random times in different random places spatio-temporal events, i.e. accidents. If the agent receives information, it adds it to its beliefs and, if the accident is on its route, it updates its intentions if possible. Agents with individual strategy have no knowledge from other agents; thus they update their route if possible once they encounter a closed route in their plan. The simulation code is available at this link: `http://modelingcommons.org/browse/one_model/4832#model_tabs_browse_info`.

3.2.4 Results and Discussion

The experiments were conducted as a version of the scenario proposed in Section 3.2.3.1, with the adoption of the three different strategies described in Section 3.2.3. The scenario is executed with 10, 50, 100, and 150 agents as part of the environment in three different experiments. We measured the time it took an agent to reach its destination. Results of the average time for agents to reach their destination for the different cases are reported in Figure 3.16. The average time for all agents to reach their destination increases as the number of agents increases. This can be explained by the traffic overload, which cannot be avoided due the number of cars on the road network. However, it is worth noticing that the time decreases when the two social agent strategies are exploited, in contrast to the individual agent strategy. Notice also that social agents using trust-based information to judge the reliability of the recommendations they receive have better performance than purely social ones. As a conclusion, the results show that exchanging spatio-temporal beliefs among agents enhances the overall performance of the agent network.

FIGURE 3.16: Average time required by the agents to reach a destination.

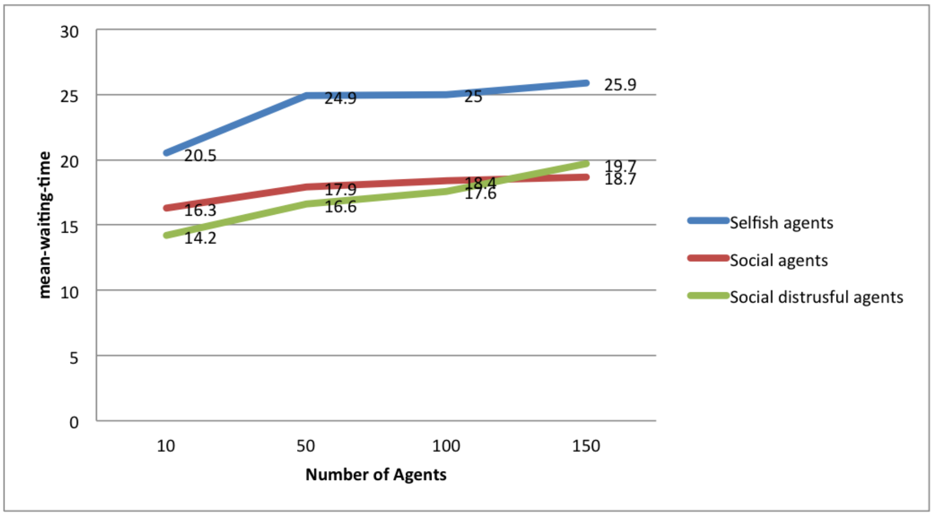


FIGURE 3.17: Average waiting time for the agents.

It is worth observing that some agents adopting the individual strategy do not even reach their destination (i.e., they cannot satisfy their goals). Therefore, the average time reported in the diagrams keeps rising indefinitely. In contrast, social agents always achieve their goals and reach their destination, with an even more limited time interval observed for those agents exploiting trust-based information. These results show that exchanging fuzzy spatio-temporal beliefs helps agents to achieve their goals by anticipating the consequences of their intentions. In other words, agents can anticipate and change their intentions to avoid huge waiting time. Taking into account spatio-temporal beliefs coming only from trustworthy agents avoids agents to be mislead and hence to waste time.

Figure 3.17 reports the average waiting time of agents. From the results it is evident that exchanging spatio-temporal beliefs among agents leads to lower waiting time for agents. Within social agents, results are slightly better for those exploiting trust-based information, except when the number of agents is 150. This is due to the time required to process such information for the whole agent network, as more processing time is needed to verify agents' reliability.

These results support the choice for agents to exchange spatio-temporal beliefs with trustworthy agents in order to achieve their spatio-temporal goals.

Chapter 4

State of the Art

Recommender systems have been proven to be valuable tools for users to cope with information overload. Originally, they were widely used on e-commerce websites [60–62] in order to guide consumers through the often-overwhelming task of identifying products they will likely to be of their interest. Lately, they have been increasingly used in the e-tourism field [63–66] providing services like recommending tourist packages from air plane tickets to activities and a lists of Points Of Interest based on users' preferences. Recently, the Artificial Intelligence community is putting much effort on the investigation and evaluation of recommender systems based on intelligent agents. Such a kind of systems has been applied so far in different fields, e.g., health-care, tourism, financial applications, or traffic and transportation (see [1, 8–10, 58, 67, 68]). In this Chapter, we carry out a literature review on recommender systems using agent and Multi-Agent systems in different domains of application. The survey discusses relevant research trends on agent-based recommender systems that have been explored so far, in particular those based on the BDI model. The advantage of such a kind of recommender systems is that the encoding of users' beliefs and goals in the system is more likely to return a recommendation as close as possible to their needs, with the possibility to include additional information like the confidence in the source. Several of the above application scenarios require to formalize the knowledge about the time and the location in which the action is taking place. This information often needs to be considered together, as in the case of the traffic scenario where a traffic jam is identified by its location and the time it is occurring during the day, and requires to encode a certain degree of vagueness as well.

4.1 Agent-based recommender systems

Combining recommender systems with agent technologies has several advantages on both sides. For agent technologies, recommender systems offer a practical and important application domain with useful concepts. For recommender systems, agent research offers ways to manage autonomy, pro-activity, distribution, reputation and trust. We are interested particularly in the use of the cognitive agent architecture as recommender system. A comparison of some proposed approaches is summarized in Table 4.1 according to different criteria.

4.1.1 Agent-based recommender systems in the tourism domain

Conventional recommendation techniques, content-based filtering [69] and collaborative filtering [70], for instance, are particularly well suited for the recommendation of products such as books, movies, or music titles. However, for products from other categories such as financial services, fitness plans or tourist packages these conventional approaches are not efficient. The reasons are manly, for example, that to recommend a tourist package, further reasoning on planning abilities is required. Furthermore, customers who use recommendation applications would not be satisfied with recommendations based on user ratings only. An alternative solution to this problem is the use of AI techniques, and particularly, agents and Multi-Agents systems.

Lately, several recommender systems in the tourism domain were proposed. Borras *et al.* [66] present in their survey of recent recommender systems in e-tourism a classification that includes agent-based recommender systems [71–75].

PersonalTour [74], a Multi-Agent recommender system, tries to reproduce a real travel agent behavior in order help customers finding the best travel packages (including flights, hotels and attractions) according to their preferences. Agents exploit knowledge about previous recommendations to determine solutions that match the customer's wishes and needs. Another interesting feature of this system is that users can give feedback about the recommended packages so that the degree of confidence on each travel agent can be updated accordingly. Although the *PersonalTour* agent gave good results compared to human travel agent, some problems related to classical recommendation are still to be faced. This problem is related to recommendations for novel users. In case of new users, the system does not have information about previous recommendations. It is not clear also how authors face problems such as novelty and serendipity within PersonalTour.

Another system focused on providing personalized services to users based on their preferences in e-tourism is presented in [72]. The Patac platform proposed in this work offers several services:

— **Personalized recommendation**: the system can provide different options for restaurants, monuments, bars, places of interest and public transport, according to the profile of the user (or the group of users) identified, her location, and the current time and weather.

— **Route planning**: it can plan an itinerary across the city (walking, by bike, by public transport or by car).

— **Social feedback**: the PaTac platform allows people to use tags, send images, add comments about places, events and services, and share all this information with other people.

Recommendations are provided by a recommender agent which makes personalized recommendations and calculates the best suggestions taking into account the user profile by means of a content-based filtering method. The originality of this work is the use of a software agent combined with Semantic Web technologies, in particular ontologies to represent users' profiles and their preferences. The system, however, does not focus on real-time or on-route update of the recommendation driven by users' change of preferences, location or some events (e.g., it starts raining).

In [76], the authors proposed SHOMAS, a Multi-Agent system that provides leisure plans for users in shopping malls. The system offers dynamic re-planning in execution time and learning from past experience thanks to the Case-Based Planning (CBP) [77] systems and Case-Based Reasoning (CBR) [78]. The CBP-CBR agent is a deliberative agent that relies on the BDI agent architecture. Although the good results, it does not solve some problems related to classical recommender systems, i.e., the cold start problem. The system needs to obtains more information about user profiles, products and habits in order to provide more optimal plans. Moreover, in this approach, there is no interaction between SHOMAS and the users, extension which may be of a high utility in this application case.

Casali *et al.* [79] presented a Travel Assistant agent that helps a tourist to choose holiday packages. They used a graded BDI agent model based on multi-context systems to deal with information uncertainty and graded notions of beliefs, desires and intentions, and a modal many-valued logic approach form modeling agents. An implementation of the proposed model is later presented in [67] and [68]. Results concluded that BDI agents are useful to build recommender systems. Nevertheless, as pointed in [80], this approach needs further research to adapt the agent behavior in a dynamic environment.

Unlike Casali *et al.* [67] where the system helps users to have appropriate leisure plans for certain destinations, Turist@ described in [1] proposes leisure activities once the user arrived at a destination according to its preferences or based on trips of similar tourists. Differently from [76], this approach combines content-based and collaborative strategies to overcome the cold start problem encountered in traditional recommender systems. The architecture of the system is shown in Figure 4.1. The core of Turist@ is the Recommender Agent, which maintains a user profile for each tourist. This profile is initialized with some basic information on high-level cultural interests provided by the user when she uses the system for the first time. The Recommender Agent dynamically and automatically refines this initial knowledge about the user preferences by analyzing the user's queries and evaluations. The Agent can also provide proactive recommendations, because it knows the position of the user in the city and can suggest cultural activities that fit the user's preferences and are located in the vicinity.

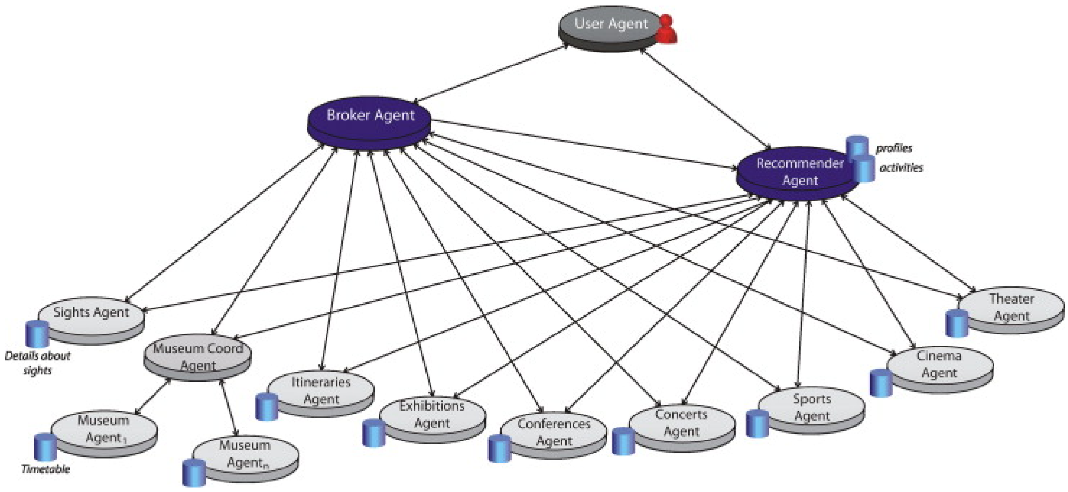


FIGURE 4.1: Architecture of Turist@ recommender system (from [1])

Real world applications, especially location-aware ones, are characterized by a lot of imprecision because of errors on localization or a lack of information. However, in this approach, there is no consideration of uncertainty when proposing personalized recommendation to users.

4.1.2 Agent-based recommender systems in the traffic field

The increasing of urban traffic jams has motivated researchers to study innovative strategies to effectively manage this problem and propose new services that fit users' requirements. Agent-based approaches have been widely investigated in traffic related problems [81] for multiple reasons. Agents provide a suitable way to model and simulate traffic systems since they offer an intuitive way to describe every autonomous entity on the individual level. Agents are reactive, they perceive their environment and respond to environmental changes. Besides, agents are collaborative, they interact and communicate with each other in order to achieve a desired goal.

Chen *et al.* [10] undertook a literature survey on agent-based approaches and their applications in the traffic and transportation domain. In their review, authors classified agent applications on traffic and transportation into five categories: *1)* agent-based traffic control and management systems; *2)* agent-based systems for roadway transportation; *3)* agent-based systems for air traffic; *4)* agent-based systems for railway transportation; and *5)* Multi-Agent traffic modelling and simulation. In this thesis, we only report about approaches related to the fifth category which is more relevant for the case study we are interested in.

Approaches regarding modeling and simulation aim principally at realistically reproducing intelligent human behaviour and decision making in scenarios that may consider high-level tasks (e.g., route choice and navigation), as well as low level ones, as for instance, driving. A number of approaches have been reported to model and implement such behavior. Bazzan *et al.* [82] propose to model the strategical level (as for instance the behaviour of drivers) in a more realistic way, at a level closer to the deliberative and social one by using mental states like beliefs, desires and intentions. This approach is detailed in the next Section. In [83], authors propose an extension to an existing microscopic simulation model called Dynamic Route Assignment Combining User Learning and microsimulAtion (DRACULA) to aid drivers' decision making. Drivers in this model are considered as cognitive entities and their behavior is handled through the use of a BDI approach, where the internal model of each agent is essentially represented by sets of beliefs, goals, and intentions.

Recent developments in agent-based modelling for traffic and transportation such as [84] combined agent-based modelling to describe the behavior of a population of cognitive agents with a macroscopic-level traffic dynamics models to constraint the movement of agents in the road network. This approach is developed to mainly face the familiar challenge of dividing computational resources between simulation volume and behavioral complexity. The hybridization of these approaches within an agent-based modelling framework yields to a representation of urban traffic flow that is driven by individual behaviour, yet, in reducing the computational intensity of the simulated physical interaction, enabling the scalable expansion to large numbers of agents.

From the literature review [10, 85], and despite the proliferation of agent-based modelling within the transportation domain, we can draw the conclusion that most of approaches fall short in adequately describing driver behaviors in a very dynamic environment which involves many individuals across a wide spatial areas that keeps changing over time. Furthermore, dealing with real-world applications rises new challenges. Problems such as imprecision and vagueness related to information coming from different resources need to be handled. More important when dealing with human behavior, uncertainty in mental attitude needs to be dealt with.

4.2 Agent-based BDI recommender systems: time and location

4.2.1 Temporal reasoning in BDI agents

Cohen and Levesque [27], and Rao and Georgeff [7] were the first to incorporate temporal components into the BDI model. The basic building block of Cohen and Levesque's BDI logic is a linear version of propositional dynamic logic (PDL). Intentions are defined in terms of temporal sequences of an agent's beliefs and goals. Each possible world extendable from a current state at a particular time point is a time line representing a sequence of events. Rao & Georgeff's approach is based on branching-time temporal logic framework to give a formal-logical definition of BDI theory. Unlike [27], instead of a time line, they choose to model the world using a temporal structure with a branching time future and a single past, called a time tree, where a particular time point in a particular world is called a situation.

Sànchez-Marrè *et al.* [86] discuss the different approaches to temporal reasoning. They classified those approaches into two main categories:

- Practical-oriented models, which are more inspired by methods such as time series models [87] and case-based reasoning [78].
- Theoretical-oriented models, which are basically inspired by logic or relation algebras. Examples include Allen's Temporal Intervals Algebra [16] and cyclic intervals by Balbiani and Osmani [88].

In [89], authors introduce a logical negotiation protocol that incorporates a real-time BDI model used to manage resource allocation problems. To incorporate real-time concerns into their logical negotiation protocol, they used several interval relationships defined by Allen [16]. To manage the negotiation between two agents, they defined several axioms that are real-time constrained thanks to functional predicates. The proposed model was applied to the distributed sensor network domain which is highly concerned by imprecision related to sensors measures. Nonetheless this constraint was not handled in the proposed solution.

So far, to the best of our knowledge, many approaches to reason about time in the BDI agent model are proposed in the literature (among them, see [90–92]) but none of them deals with time information imprecision.

4.2.2 Spatial reasoning in BDI agents

Schuele *et al.* [93] propose a spatial model to enable BDI agents to move autonomously and collision-free in a spatial environment. Spatial reasoning is handled in this approach through the RCC-8 qualitative relations in a GISAgent component. The GIS agent consists of a BDIAgent with its SpatialRepresentation and SpatialReasoner (that implements the RCC-8 spatial relations) and a GISLibrary containing spatial data compliant with the OpenGIS standard. Authors assume that in a spatial context, the agents' knowledge about their environment is uncertain. However, this problem is not handled through a qualitative approach for spatial reasoning. Time reasoning is not handled neither.

SISMORA introduced in [2] proposes an architecture that combines Multi-Agent systems with GIS in which multiple moving agents collaborate in a geo-spatial environment in order to achieve a goal. Qualitative and declarative relationships in terms of distance, direction, topology and motion are included in this architecture as an axiomatic first-order-logic system. Agent decision making is based on a GIS-based Belief-Desire-Intention model visualized in Figure 4.2. The difference with a traditional BDI model is the fact that a belief can be a spatial and motion fact retrieved from GIS geo-databases, desires contain a plan of actions based on an agent's goal in a geo-spatial space.

Issues such as information consistency, i.e., introducing new beliefs or desires that do not contradict existing ones, and temporal reasoning in order to simulate real situations are still open issues in this approach.

Other relevant approaches for spatial reasoning within Multi-Agents systems are discussed in [86]. Authors pointed out in their analysis that simultaneous reasoning in space and time is difficult to handle and requires a lot of computing resources, and the fact of using autonomous agents limits this overcome. Another open issue pointed in their review is the need to handle uncertainty because, according to the authors, "*as soon as a real-life system is studied and analyzed, uncertainty is indeed inherently present*". However, they do not consider in their analysis the imprecision and vagueness coupled to spatial knowledge.

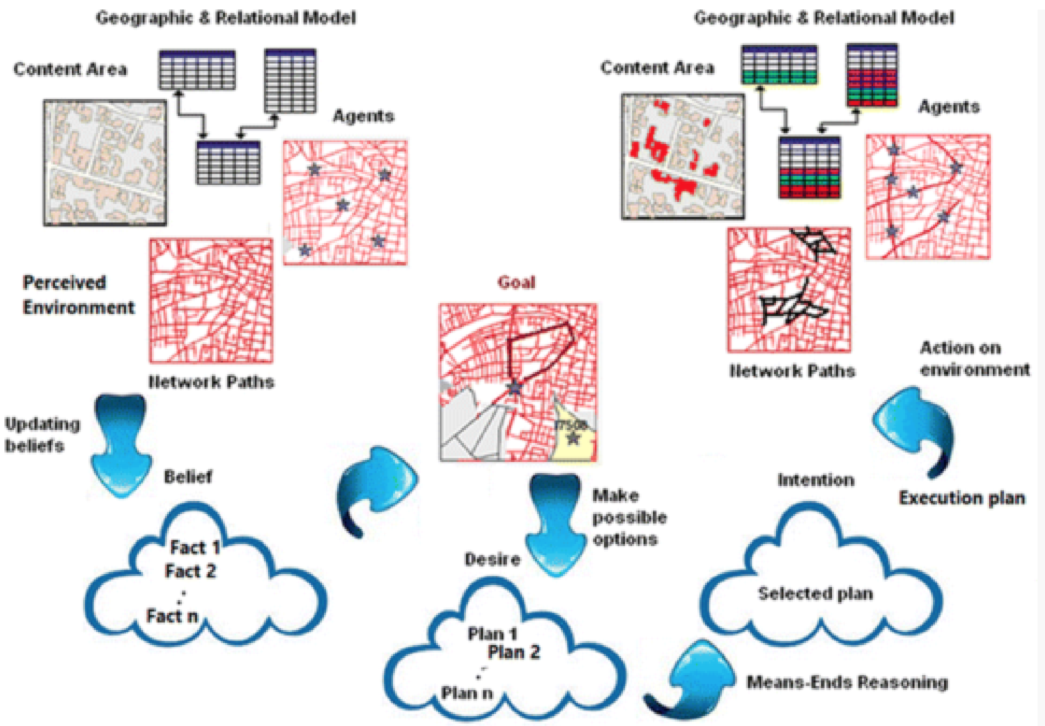


FIGURE 4.2: The SISMORA's GIS-based BDI Model (from [2])

4.2.3 Spatio-temporal BDI systems dynamics

Few approaches exist to represent and reason about spatio-temporal beliefs, desires and intentions' dynamics. Jonker *et al.* [94] propose a formal spatio-temporal state language with the aim to define the spatio-temporal behavior of an agent in a dynamic environment. Although their approach provides an interesting formalism for predicting agent spatial behavior, many questions concerning beliefs, desires and intentions dynamics are left open. For example, no mechanism for updating beliefs, desires and intentions in this formalism is presented.

Maleš and colleagues [95] use modal logic to define an agent capable of updating its mental attitude according to spatio-temporal relations considered as events. They define a language for events in which spatio-temporal knowledge is defined under the form of predicates. An example of a pedestrian in a traffic scenario was presented and implemented in NetLogo. Results show the usefulness of this model in a simple real-world scenario. Nevertheless, the proposed framework still is in a preliminary stage and presents some drawbacks, e.g., a mechanism to update such spatio-temporal beliefs and desires is missing.

Unlike the aforementioned approaches, our approach besides combining spatial and temporal reasoning within the BDI model, it addresses the open challenge of spatio-temporal information vagueness and fuzziness that strongly characterizes such a kind of knowledge.

4.3 Uncertainty Management in MAS

The use of MAS provides a clear added value in autonomy compared to conventional systems. However, MAS have a limited ability to deal with uncertainty in a dynamic environments. Uncertainty can arise from many factors, such as complexity, randomness, ignorance, or imprecision.

Zadeh [32] states that "complexity and precision are incompatible properties", arguing that the conventional approaches are inadequate to model human-like complex processes. Therefore, "the closer one looks at a real-world problem, the fuzzier becomes its solution". Fuzzy set theory (introduced in Chapter 2 Section 2.3.1) is widely used in the Artificial Intelligence field to deal with uncertain problem domain. Agents, that implement uncertain problems by means of fuzzy logic, are called fuzzy agents. Fuzzy agents are used in fuzzy reasoning situations, where agents interpret a situation, solve a problem or decide with respect to the available fuzzy knowledge [96–98].

In the BDI architecture, few approaches handle this issue. Among them, the approach proposed in [99]. It presents a BDI agent model with fuzzy perception used as personal assistants for giving personalized recommendations to individual on-line users in a used car electronic market over the Internet. Fuzzy agents are defined via extended fuzzy cognitive maps. Long and Esterline [100] introduce a BDI agent, which uses fuzzy inference engines, fuzzy controllers and classifiers, for the modeling of co-operative societies of artificial agents. Shen *et al.* [101] have explored a hybrid BDI model based on deliberative and fuzzy reasoning, and they improved the model in [102] within the context of wireless sensor networks.

Casali *et al.*[79, 80, 103] proposed to handle uncertainty through possibility theory by defining graded mental attitude, i.e., the degree to which an agent believes that a formula is true. However this is different from handling information imprecision related to real-world constraints, e.g., errors in sensor measures.

Approaches \ Features	Agent-based	Social	Trust	Uncertainty	Spatial Reasoning	Temporal Reasoning
Casali *et al.* [79]	√	√	√	√	X	X
Bajo *et al.* [76]	√	X	X	X	√	√
Batet *et al.* [1]	√	X	X	X	√	X
Lorenzi *et al.* [74]	√	X	X	X	X	X
Ceccaroni *et al.* [72]	√	X	X	X	X	X
Rossetti *et al.* [83]	√	X	X	X	√	X
Manley *et al.* [84]	√	X	X	X	√	X
Vahidnia *et al.* [2]	√	X	X	X	√	√
Maleš *et al.* [95]	√	X	X	X	√	√
Miao *et al.* [104]	√	X	X	√	√	√
Othmane *et al.* [105]	√	√	√	√	√	√

Table 4.1: A comparison of different agent-based recommendation approaches according to their main features.

Chapter 5

Conclusion

This chapter summarizes the main contributions of this thesis, draws some conclusions, reviews some of the research issues that remain to be addressed by pointing out some promising directions for future work.

5.1 Contributions

This thesis investigated agent-based recommender systems as an efficient way for decision-making motivated by the open challenges in terms of customization, re-activity and autonomy raised by real-world applications. Our research was motivated by the following research questions:

RQ1: *how to define a recommender system able to deal with the flexibility, complexity, uncertainty and dynamics required for real world applications?*

To answer this question, we proposed to use Multi-Agents systems as a recommender system to customize recommendations. The resulting framework, called **CARS** (Cognitive Agent-based Recommender System), is based on the agent technology to enhance the recommender system with the cognition, social ability and the autonomy required in a dynamic environment. MAS are characterized by a high degree of uncertainty since they are composed by heterogeneous agents acting in a dynamic environment. For this reason, a solution to handle the degree of belief in an information is to use possibility theory. In simulated scenarios, experiments show that agents achieve a better performance collectively when they are part of "communities", i.e., agents exchange messages with the other agents with shared interests. In addition, we also studied the impact of trust on the received recommendations. Results show that exchanging beliefs and desires with trustworthy agents can ameliorate the whole performance of the agents.

RQ2: *how to represent and reason about fuzzy spatial-temporal knowledge to provide useful recommendations?*

This research question has been addressed through the definition of an extension of **CARS**, enhanced with fuzzy spatio-temporal reasoning. In order to represent fuzzy spatio-temporal information to provide recommendations, we defined spatio-temporal knowledge annotating spatial formulae (formalized through fuzzy RCC) with temporal information (formalized through fuzzy Allen's time intervals). The goodness of the proposed formal framework is validated through an empirical evaluation simulating the agents' behaviour in the traffic scenario. Results show that the time required by the agents to reach a certain point of interest sensibly decreases when the CARS model is applied.

We believe that agents are a good alternative to traditional recommendation techniques in designing real-world applications thanks to their social dimension, cognitive abilities, autonomy and reactivity. BDI agents, in particular, are well suited in applications that involve humans when the decision-making process is driven not only by rational thinking but by some emotional components such as beliefs and desires as well. BDI agents with fuzzy perception seem to be a good model to be used in agent-based simulations in environments with imperfect information.

5.2 Perspectives

We list here some directions that are considered to extend the research presented in this thesis. First of all, further qualitative relations about directions should be introduced concerning spatial reasoning to allow the representation of a model closer to reality. Second, on the simulation side, extending the evaluation introducing new metrics to further reduce the processing time and compare the performance of the system with different strategies is to be considered. Third, an empirical evaluation of the planning module in particular is ongoing with the aim to study the advantages of using the proposed ontology compared to traditional planning methods.

Lastly, we discuss some directions for future work in the context of MAS, and more precisely, of BDI agent systems:

— According to Berners-Lee *et al.* [106], *the actual influence of semantic technology will be uncovered when people are able to develop intelligent agents capable of acquiring knowledge from different sources, manipulating them and sharing the results amongst them. Utilizing the power underpinning semantic technology, agents are able to perform the entire aforementioned activities automatically.* In order to meet this vision, two key technologies are identified : agents for representing real-world entities and automated task resolution, and ontologies for semantically enhanced information exchange and processing over the Web. These two technologies need to be integrated in a coherent framework especially for the domains where relevant information is widely distributed. In fact, Semantic Web technologies have proved to be very useful in solving the heterogeneity problem. They offer a common framework that enables for data integration, sharing and reuse from multiple sources.

— Emotions including moods, feelings, and personality have a strong effect on peoples' physical states, motivations, beliefs, and desires. However, they are often not taken into account in designing and implementing BDI models. Integrating emotions such as fear, self-confidence or happiness in the reasoning and decision making process of BDI agents can be more representative of human behavior, allowing for the combination of practical rational elements with more "human-based" features in agent reasoning. Approaches such as those proposed by Pereira *et al.* [107] and Jiang *et al.* [108] for emotional BDI frameworks can be explored.

5.3 Publications

Published papers

- Amel Ben Othmane, Andrea Tettamanzi, Serena Villata, Nhan Le Thanh, and Michel Buffa. A multi-context framework for modeling an agent-based recommender system. In H. Jaap van den Herik and Joaquim Filipe, editors, *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016), Volume 2, Rome, Italy, February 24-26, 2016.*, pages 31–41. SciTePress, 2016. doi: 10.5220/0005686500310041. URL http://dx.doi.org/10.5220/0005686500310041
- Amel Ben Othmane, Andrea Tettamanzi, Serena Villata, Nhan Le Thanh, and Michel Buffa. An agent-based architecture for personalized recommendations. In H. Jaap van den Herik and Joaquim Filipe, editors, *Agents and Artificial Intelligence - 8th International Conference, ICAART 2016, Rome, Italy, February 24-26, 2016, Revised Selected Papers*, volume 10162 of *Lecture Notes in Computer Science*, pages 96–113, 2016. ISBN 978-3-319-53353-7. doi: 10.1007/978-3-319-53354-4_6. URL http://dx.doi.org/10.1007/978-3-319-53354-4_6
- Amel Ben Othmane, Andrea Tettamanzi, Serena Villata, and Nhan Le Thanh. A multi-context BDI recommender system: From theory to simulation. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA, October 13-16, 2016*, pages 602–605, 2016. doi: 10.1109/WI.2016.0104. URL http://dx.doi.org/10.1109/WI.2016.0104
- Amel Ben Othmane, Andrea G. B. Tettamanzi, Serena Villata, and Nhan Le Thanh. Towards a spatio-temporal agent-based recommender system. In *16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), Das, Durfee, Larson, Winikoff (eds.), May 8-12, Sao Paulo, Brazil.*, 2017

Papers under review

- Amel Ben Othmane, Andrea G. B. Tettamanzi, Serena Villata, and Nhan Le Thanh. CARS – a spatio-temporal BDI recommender system: Time, space and uncertainty. In *11th International Conference on Scalable Uncertainty Management - Granada, Spain, October 4-6*, 2017
- Amel Ben Othmane, Andrea G. B. Tettamanzi, Serena Villata, and Nhan Le Thanh. Cars – an agent-based recommender system: Formal framework and empirical evaluation. In *International Journal on Artificial Intelligence Tools. World Scientific. Under review*, 2017

Bibliography

[1] Montserrat Batet, Antonio Moreno, David Sánchez, David Isern, and Aïda Valls. Turist@: Agent-based personalised recommendation of tourist activities. *Expert Systems with Applications*, 39(8):7319–7329, 2012.

[2] Mohammad H Vahidnia, Ali A Alesheikh, and Seyed Kazem Alavipanah. A multi-agent architecture for geosimulation of moving agents. *Journal of Geographical Systems*, 17(4):353–390, 2015.

[3] Steven Schockaert, Martine De Cock, and Etienne E Kerre. Spatial reasoning in a fuzzy region connection calculus. *Artificial Intelligence*, 173(2):258–298, 2009.

[4] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer, 2011. ISBN 978-0-387-85819-7. URL http://dblp.uni-trier.de/db/reference/rsh/rsh2011.html#LopsGS11.

[5] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. Collaborative filtering recommender systems. *Found. Trends Hum.-Comput. Interact.*, 4(2): 81–173, February 2011. ISSN 1551-3955. doi: 10.1561/1100000009. URL http://dx.doi.org/10.1561/1100000009.

[6] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002. ISSN 0924-1868. doi: 10.1023/A:1021240730564. URL http://dx.doi.org/10.1023/A:1021240730564.

[7] Anand S Rao, Michael P Georgeff, et al. Bdi agents: From theory to practice. In *ICMAS*, volume 95, pages 312–319, 1995.

[8] Beatriz Lopez, Silvana Aciar, Bianca Innocenti, and Isabel Cuevas. How multi-agent systems support acute stroke emergency treatment. In *IJCAI Workshop on Agents Applied in Health Care*, pages 51–59, 2005.

[9] Longbing Cao and Chengqi Zhang. F-trade: an agent-mining symbiont for financial services. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 262. ACM, 2007.

[10] Bo Chen and Harry H Cheng. A review of the applications of agent technology in traffic and transportation systems. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):485–497, 2010.

[11] Miquel Montaner, Beatriz López, and Josep Lluís De La Rosa. A taxonomy of recommender agents on the internet. *Artificial intelligence review*, 19(4):285–330, 2003.

[12] Célia da Costa Pereira and Andrea G. B. Tettamanzi. An integrated possibilistic framework for goal generation in cognitive agents. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 1239–1246, 2010.

[13] Célia da Costa Pereira and Andrea G. B. Tettamanzi. Syntactic possibilistic goal generation. In *ECAI 2014-21st European Conference on Artificial Intelligence*, volume 263, pages 711–716. IOS Press, 2014.

[14] Simon Parsons, Nicholas R Jennings, Jordi Sabater, and Carles Sierra. Agent specification using multi-context systems. In *Foundations and Applications of Multi-Agent Systems*, pages 205–226. Springer, 2002.

[15] Lotfi A Zadeh. The concept of a linguistic variable and its application to approximate reasoning—i. *Information sciences*, 8(3):199–249, 1975.

[16] James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[17] David A Randell, Zhan Cui, and Anthony G Cohn. A spatial logic based on regions and connection. *KR*, 92:165–176, 1992.

[18] Jacques Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading, 1999.

[19] Michael Wooldridge and Nicholas R Jennings. Agent theories, architectures, and languages: a survey. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 1–39. Springer, 1994.

[20] Pattie Maes. *Designing autonomous agents: Theory and practice from biology to engineering and back.* MIT press, 1990.

[21] Allen Newell and Herbert A Simon. Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3):113–126, 1976.

[22] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[23] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02):115–152, 1995.

[24] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE journal on robotics and automation*, 2(1):14–23, 1986.

[25] Innes A Ferguson. Touring machines: Autonomous agents with attitudes. *Computer*, 25(5):51–55, 1992.

[26] Michael Bratman. Intention, plans, and practical reason. 1987.

[27] Philip R Cohen and Hector J Levesque. Intention is choice with commitment. *Artificial intelligence*, 42(2):213–261, 1990.

[28] Michael Wooldridge, Nicholas R Jennings, and David Kinny. The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and multi-agent systems*, 3(3):285–312, 2000.

[29] Munindar P Singh. Semantical considerations on intention dynamics for bdi agents. *Journal of Experimental & Theoretical Artificial Intelligence*, 10(4):551–564, 1998.

[30] Gerhard Brewka and Thomas Eiter. Equilibria in heterogeneous nonmonotonic multi-context systems. In *AAAI*, volume 7, pages 385–390, 2007.

[31] Fausto Giunchiglia and Luciano Serafini. Multilanguage hierarchical logics, or: how we can do without modal logics. *Artificial intelligence*, 65(1):29–70, 1994.

[32] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[33] Glad Deschrijver, Chris Cornelis, and Etienne E Kerre. On the representation of intuitionistic fuzzy t-norms and t-conorms. *IEEE transactions on fuzzy systems*, 12(1):45–61, 2004.

[34] M. Navara. Triangular norms and conorms. *Scholarpedia*, 2(3):2398, 2007. revision #137537.

[35] C Negoita, L Zadeh, and H Zimmermann. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1:3–28, 1978.

[36] Franziska Klügl and Ana LC Bazzan. Agent-based modeling and simulation. *AI Magazine*, 33(3):29, 2012.

[37] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3): 7280–7287, 2002.

[38] Frank Lui, Dennis Jarvis, and Russell Connell. An architecture to support autonomous command agents for onesaf testbed simulations. In *Proceedings of the SimTecT Conference 2002*. Citeseer, 2002.

[39] Federico Cecconi, Domenico Parisi, et al. Individual versus social survival strategies. *Journal of Artificial Societies and Social Simulation*, 1(2):1–17, 1998.

[40] Ameya Shendarkar, Karthik Vasudevan, Seungho Lee, and Young-Jun Son. Crowd simulation for emergency response using bdi agents based on immersive virtual reality. *Simulation Modelling Practice and Theory*, 16(9):1415–1429, 2008.

[41] Charles M Macal and Michael J North. Tutorial on agent-based modelling and simulation. *Journal of simulation*, 4(3):151–162, 2010.

[42] Carole Adam and Benoit Gaudou. Bdi agents in social simulations: a survey. *The Knowledge Engineering Review*, 31(3):207–238, 2016.

[43] Steven F Railsback, Steven L Lytinen, and Stephen K Jackson. Agent-based simulation platforms: Review and development recommendations. *Simulation*, 82 (9):609–623, 2006.

[44] Steven L Lytinen and Steven F Railsback. The evolution of agent-based simulation platforms: a review of netlogo 5.0 and relogo. In *Proceedings of the fourth international symposium on agent-based modeling and simulation*, 2012.

[45] Nelson Minar, Roger Burkhart, Chris Langton, Manor Askenazi, et al. The swarm simulation system: A toolkit for building multi-agent simulations. Santa Fe Institute Santa Fe, 1996.

[46] Nick Collier. Repast: An extensible framework for agent simulation. *The University of Chicago's Social Science Research*, 36:371–375, 2003.

[47] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, 2005.

[48] U Wilensky. Netlogo. *Center for Connected Learning and Computer-Based Modelling, Northwestern University*, 2012.

[49] Didier Dubois and Henri Prade. *Possibility theory and its applications: a retrospective and prospective view.* Springer, 2006.

[50] Fabio Paglieri, Cristiano Castelfranchi, Célia da Costa Pereira, Rino Falcone, Andrea Tettamanzi, and Serena Villata. Trusting the messenger because of the message: feedback dynamics from information quality to source evaluation. *Computational and Mathematical Organization Theory*, 20(2):176–194, 2014.

[51] Patricia Victor, Chris Cornelis, Martine De Cock, and Paulo Pinheiro Da Silva. Gradual trust and distrust in recommender systems. *Fuzzy Sets and Systems*, 160 (10):1367–1382, 2009.

[52] Tom Gruber. Ontology. *Encyclopedia of database systems*, pages 1963–1965, 2009.

[53] Ilias Sakellariou, Petros Kefalas, and Ioanna Stamatopoulou. Enhancing netlogo to simulate bdi communicating agents. In *Artificial Intelligence: Theories, Models and Applications*, pages 263–275. Springer, 2008.

[54] Jan Thiele. R marries netlogo: Introduction to the rnetlogo package. *Journal of Statistical Software*, 58(1), 2014.

[55] Antony Galton. Spatial and temporal knowledge representation. *Earth Science Informatics*, 2(3):169–187, 2009.

[56] Gloria Bordogna, Paola Carrara, Sergio Chiesa, and Stefano Spaccapietra. A dual representation of uncertain dynamic spatial information. In *International Fuzzy Systems Association World Congress*, pages 652–659. Springer, 2003.

[57] Steven Schockaert, Chris Cornelis, Martine De Cock, and Etienne E Kerre. Fuzzy spatial relations between vague regions. In *2006 3rd International IEEE Conference Intelligent Systems*, pages 221–226. IEEE, 2006.

[58] Amel Ben Othmane, Andrea Tettamanzi, Serena Villata, Nhan Le Thanh, and Michel Buffa. An agent-based architecture for personalized recommendations. In H. Jaap van den Herik and Joaquim Filipe, editors, *Agents and Artificial Intelligence - 8th International Conference, ICAART 2016, Rome, Italy, February 24-26, 2016, Revised Selected Papers*, volume 10162 of *Lecture Notes in Computer Science*, pages 96–113, 2016. ISBN 978-3-319-53353-7. doi: 10.1007/978-3-319-53354-4_6. URL http://dx.doi.org/10.1007/978-3-319-53354-4_6.

[59] Amel Ben Othmane, Andrea Tettamanzi, Serena Villata, and Nhan Le Thanh. A multi-context BDI recommender system: From theory to simulation. In *2016*

IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA, October 13-16, 2016, pages 602–605, 2016. doi: 10.1109/WI.2016.0104. URL http://dx.doi.org/10.1109/WI.2016.0104.

[60] J Ben Schafer, Joseph A Konstan, and John Riedl. E-commerce recommendation applications. In *Applications of Data Mining to Electronic Commerce*, pages 115–153. Springer, 2001.

[61] Kangning Wei, Jinghua Huang, and Shaohong Fu. A survey of e-commerce recommender systems. In *Service systems and service management, 2007 international conference on*, pages 1–5. IEEE, 2007.

[62] Jan Boehmer, Yumi Jung, and Rick Wash. e-commerce recommender systems. *The International Encyclopedia of Digital Communication and Society*, 2015.

[63] Markus Zanker, Matthias Fuchs, Wolfram Höpken, Mario Tuta, and Nina Müller. Evaluating recommender systems in tourism—a case study from austria. *Information and communication technologies in tourism 2008*, pages 24–34, 2008.

[64] Helmut Berger, Michael Dittenbach, Dieter Merkl, Anton Bogdanovych, Simeon Simoff, and Carles Sierra. Opening new dimensions for e-tourism. *Virtual Reality*, 11(2-3):75–87, 2007.

[65] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, and Grammati Pantziou. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39:319–333, 2014.

[66] Joan Borras, Antonio Moreno, and Aida Valls. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16):7370–7389, 2014.

[67] Ana Casali, Lluís Godo, and Carles Sierra. A tourism recommender agent: from theory to practice. *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, 12(40):23–38, 2008.

[68] Ana Casali, Lluís Godo, and Carles Sierra. Validation and experimentation of a tourism recommender agent based on a graded bdi model. In *CCIA*, pages 41–50, 2008.

[69] Robin Van Meteren and Maarten Van Someren. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, pages 47–56, 2000.

[70] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[71] Luis Castillo, Eva Armengol, Eva Onaindía, Laura Sebastiá, Jesús González-Boticario, Antonio Rodríguez, Susana Fernández, Juan D Arias, and Daniel Borrajo. Samap: An user-oriented adaptive system for planning tourist visits. *Expert Systems with Applications*, 34(2):1318–1332, 2008.

[72] Luigi Ceccaroni, Victor Codina, Manel Palau, and Marc Pous. Patac: Urban, ubiquitous, personalized services for citizens and tourists. In *Digital Society, 2009. ICDS'09. Third International Conference on*, pages 7–12. IEEE, 2009.

[73] Chang-Shing Lee, Young-Chung Chang, and Mei-Hui Wang. Ontological recommendation multi-agent for tainan city travel. *Expert Systems with Applications*, 36(3):6740–6753, 2009.

[74] Fabiana Lorenzi, Stanley Loh, and Mara Abel. Personaltour: a recommender system for travel packages. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02*, pages 333–336. IEEE Computer Society, 2011.

[75] Laura Sebastia, Adriana Giret, and Inma Garcia. A multi agent architecture for tourism recommendation. In *Trends in Practical Applications of Agents and Multiagent Systems*, pages 547–554. Springer, 2010.

[76] Javier Bajo, Juan M Corchado, Yanira De Paz, Juan F De Paz, Sara Rodríguez, Quintín Martín, and Ajith Abraham. Shomas: Intelligent guidance and suggestions in shopping centres. *Applied Soft Computing*, 9(2):851–862, 2009.

[77] Luca Spalzzi. A survey on case-based planning. *Artificial Intelligence Review*, 16 (1):3–36, 2001.

[78] Janet L Kolodner. An introduction to case-based reasoning. *Artificial intelligence review*, 6(1):3–34, 1992.

[79] Ana Casali, Lluís Godo, and Carles Sierra. Modeling travel assistant agents: a graded bdi approach. In *Artificial Intelligence in Theory and Practice*, pages 415–424. Springer, 2006.

[80] Ana Casali, Lluís Godo, and Carles Sierra. A graded bdi agent model to represent and reason about preferences. *Artificial Intelligence*, 175(7):1468–1478, 2011.

[81] Franziska Klügl, Ana Bazzan, and Sascha Ossowski. *Applications of agent technology in traffic and transportation.* Springer Science & Business Media, 2006.

[82] Ana Lécia C Bazzan, Joachim Wahle, and Franziska Klügl. Agents in traffic modelling—from reactive to social behaviour. In *Annual Conference on Artificial Intelligence*, pages 303–306. Springer, 1999.

[83] Rosaldo JF Rossetti, Rafael H Bordini, Ana LC Bazzan, Sergio Bampi, Ronghui Liu, and Dirck Van Vliet. Using bdi agents to improve driver modelling in a commuter scenario. *Transportation Research Part C: Emerging Technologies*, 10 (5):373–398, 2002.

[84] Ed Manley, Tao Cheng, Alan Penn, and Andy Emmonds. A framework for simulating large-scale complex urban traffic dynamics through hybrid agent-based modelling. *Computers, Environment and Urban Systems*, 44:27–36, 2014.

[85] Ana LC Bazzan and Franziska Klügl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(03):375–403, 2014.

[86] Miquel Sànchez-Marrè, Karina Gibert, Rick Sojda, Jean-Philippe Steyer, Peter Struss, and Ignasi Rodríguez-Roda. *Uncertainty management, spatial and temporal reasoning, and validation of intelligent environmental decision support systems.* PhD thesis, International Environmental Modelling and Software Society, 2006.

[87] Brian Knight and Jixin Ma. Time representation: A taxonomy of temporal models. *Artificial Intelligence Review*, 7(6):401–419, 1993.

[88] Philippe Balbiani and Aomar Osmani. A model for reasoning about topologic relations between cyclic intervals. In *KR*, pages 378–385, 2000.

[89] Leen-Kiat Soh and Costas Tsatsoulis. A real-time negotiation model and a multi-agent sensor network implementation. *Autonomous Agents and Multi-Agent Systems*, 11(3):215–271, 2005.

[90] Bevan Jarvis, Dan Corbett, and Lakhmi C Jain. Reasoning about time in a bdi architecture. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 851–857. Springer, 2005.

[91] Michael Fisher. Temporal development methods for agent-based. *Autonomous Agents and Multi-Agent Systems*, 10(1):41–66, 2005.

[92] Carles Sierra and Liz Sonenberg. A real-time negotiation model and a multi-agent sensor network implementation. *Autonomous Agents and Multi-Agent Systems*, 11 (1):5–6, 2005.

[93] Michael Schuele and Paul Karaenke. Qualitative spatial reasoning with topological information in bdi agents. In *Proceedings of the 2nd Workshop on Artificial Intelligence and Logistics (AILog), Lisbon, Portugal*, pages 7–12, 2010.

[94] Catholijn M Jonker, Vagan Terziyan, and Jan Treur. Temporal and spatial analysis to personalise an agent's dynamic belief, desire, and intention profiles. In *Cooperative Information Agents VII*, pages 298–315. Springer, 2003.

[95] Lada Maleš and Berislav Žarnić. A spatiotemporal model of events within a bdi. In *EUROCON-International Conference on Computer as a Tool (EUROCON), 2011 IEEE*, pages 1–4. IEEE, 2011.

[96] AF Gómez Skarmeta, H Martínez Barberá, and M Sánchez Alonso. A fuzzy agents architecture for autonomous mobile robots. In *Procs of the World Congress of the Int Fuzzy Systems Association (IFSA)*, 1999.

[97] Faiyaz Doctor, Hani Hagras, and Victor Callaghan. A type-2 fuzzy embedded agent to realise ambient intelligence in ubiquitous computing environments. *Information Sciences*, 171(4):309–334, 2005.

[98] Samer Hassan, Luis Garmendia, and Juan Pavón. Agent-based social modeling and simulation with fuzzy sets. In *Innovations in Hybrid Intelligent Systems*, pages 40–47. Springer, 2007.

[99] Giovani Parente Farias, Graçaliz Pereira Dimuro, and Antônio Carlos da Rocha Costa. Bdi agents with fuzzy perception for simulating decision making in environments with imperfect information. In *MALLOW*, 2010.

[100] Stephen A Long and Albert C Esterline. Fuzzy bdi architecture for social agents. In *Southeastcon 2000. Proceedings of the IEEE*, pages 68–74. IEEE, 2000.

[101] Song Shen, Gregory MP O'Hare, and Rem Collier. Decision-making of bdi agents, a fuzzy approach. In *Computer and Information Technology, 2004. CIT'04. The Fourth International Conference on*, pages 1022–1027. IEEE, 2004.

[102] Song Shen, G MP O'Hare, and Michael J O'Grady. Fuzzy-set-based decision making through energy-aware and utility agents within wireless sensor networks. *Artificial Intelligence Review*, 27(2):165–187, 2007.

[103] Ana Casali et al. *On intentional and social agents with graded attitudes.* Universitat de Girona, 2008.

[104] Chunyan Miao, Qiang Yang, Haijing Fang, and Angela Goh. A cognitive approach for agent-based personalized recommendation. *Knowledge-Based Systems*, 20(4): 397–405, 2007.

[105] Amel Ben Othmane, Andrea G. B. Tettamanzi, Serena Villata, and Nhan Le Thanh. Towards a spatio-temporal agent-based recommender system. In *16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), Das, Durfee, Larson, Winikoff (eds.), May 8-12, Sao Paulo, Brazil.*, 2017.

[106] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[107] David Pereira, Eugénio Oliveira, and Nelma Moreira. Formal modelling of emotions in BDI agents. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 62–81. Springer, 2007.

[108] Hong Jiang, Jose M Vidal, and Michael N Huhns. Ebdi: an architecture for emotional agents. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 11. ACM, 2007.

[109] Amel Ben Othmane, Andrea Tettamanzi, Serena Villata, Nhan Le Thanh, and Michel Buffa. A multi-context framework for modeling an agent-based recommender system. In H. Jaap van den Herik and Joaquim Filipe, editors, *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016), Volume 2, Rome, Italy, February 24-26, 2016.*, pages 31–41. SciTePress, 2016. doi: 10.5220/0005686500310041. URL http://dx.doi.org/10.5220/0005686500310041.

[110] Amel Ben Othmane, Andrea G. B. Tettamanzi, Serena Villata, and Nhan Le Thanh. CARS – a spatio-temporal BDI recommender system: Time, space and uncertainty. In *11th International Conference on Scalable Uncertainty Management - Granada, Spain, October 4-6*, 2017.

[111] Amel Ben Othmane, Andrea G. B. Tettamanzi, Serena Villata, and Nhan Le Thanh. Cars – an agent-based recommender system: Formal framework and empirical evaluation. In *International Journal on Artificial Intelligence Tools. World Scientific. Under review*, 2017.