Institute of Statistical Studies and Research

Cairo University

A Framework for Agent Collaboration in Multi-Agent Systems

Submitted by

Mohamed Gamaleldin Atwany

Supervised by

Abdel-Aziz Khamis, Phd.

Dept. of Computer and Information Sciences,

Cairo University

Magdy Aboul-Ela, Phd.

Dept. of Computer and Information Systems,

Sadat Academy for Management Sciences

A thesis submitted to the Department of Computer Science, Institute of Statistical Studies and Research, Cairo University, in partial fulfillment of the requirements for the degree of Master in Computer Science

May 2002

STATEMENT

I CERTIFY THAT THIS WORK HAS NOT BEEN ACCEPTED IN SUBSTANCE OF ANY ACADEMIC DEGREE AND IS NOT BEING CONCURRENTLY SUBMITTED IN CANDIDATURE FOR ANY OTHER DEGREE. ANY PORTION OF THIS THESIS FOR WHICH I AM INDEBTED TO OTHER SOURCES IS MENTIONED AND EXPLICIT REFERENCES ARE GIVEN.

STUDENT NAME: MOHAMED GAMALELDIN ATWANY

SIGNATURE:

To My Parents,

To My Wife Noha,

To My Children Abdel Rahman and Ali

For their Support and Care

Acknowledgement

This work wouldn’t come to light without the help and support of many people. Special thanks to the efforts of Professor Abdel Aziz Khamis and Professor Magdy Abul-Ela. Also, I would like to thank Professor Mervat Gheith for directing me to research in field of Artificial Intelligence. I would also take this opportunity to thank Professor Atef Abdel Moneam, with whom I have attended four classes on mathematics.

I would also like to thank my colleagues who joined me during the pre-master courses, for their support and kindness. I would give special thanks to Mahinaz Shawki, Suzan Hussain, Mohamed Moustafa, Abdullah Mohamed Abdullah and to many others.

For my little family, I cannot forget the unbelievable support of my wife and sons Abdel Rahman and Ali. I also give my deepest gratitude for my mother and father, and my wife’s mother and father who did not forget me in their *Doaa*.

Abstract

In this research, we introduce the development of a framework that supports collaboration between a group of heterogeneous agents, supporting the development and maintenance of mutual understanding and a shared view of the task being solved. The developed framework builds on two well known formal models of teamwork namely, the Joint Intentions Theory proposed by P. Cohen and H. Levesque and the Cooperative Problem Solving Model by M. Wooldridge and N. Jennings, and builds on layered agent conversational model defined by M. Nowostawski, M. Purvis and S. Cranefield.

Rather than assuming specific agent architecture, the developed collaboration framework develops a set of specifications that an agent should follow in order to guarantee its success as a team member. In this sense, the proposed approach focuses on creating a high-level collaboration protocol for developing and maintaining a mutual understanding and a shared view of teamwork. In addition, the developed framework is transparent to agent architecture, organization and interaction protocols, enabling it to be used in different domains and on several platforms, and allowing integration with various agent development environments, and allowing interoperability between agents having different agent architectures.

The developed framework maintains a clear separation between the conceptual model, defined once, and the possible model implementation, where implementations introduce details of messaging, ontology, communication and interaction. In this research, we propose and implement a framework that we utilize in the development and implementation of the case study, which is the problem of performing trade exchange and settlement between a team of trading agents.

Table of Content

Chapter 1
Introduction to Multi-Agent Systems

In this chapter, we introduce the concept of multi-agent systems. In section 1.1, we define agents and multi-agent systems. In sections 1.2 through 1.7, we discuss a number of key issues in the domain of multi-agent systems, enumerated as communication, interactions, coordination, cooperation, and organizations. Finally, in section 1.8, we introduce one of the major application areas of multi-agent systems, named distributed problem solving.

1.1 Definitions

In this research, we adopt the following definitions for an agency and multi-agent systems.

An agent is a virtual or physical computational entity that has the following characteristics [1, 2]:

- An agent can be viewed as perceiving and acting upon its environment, which it might have partial representation of.
- An agent may be able to reproduce itself.
- An agent can communicate directly with other agents.
- An agent possesses skills and can offer services.
- An agent possesses resources of its own.
- An agent is driven by a set of tendencies in the form of individual objectives or of satisfaction/survival function which it tries to optimize.
- Agents are autonomous in the sense that its behavior tends towards satisfying its objectives, taking account of the resources and skills available to it and depending on its perception, its representation and the communications it receives.
- Behavioral flexibility and rationality are achieved by an agent on the basis of key processes such as problem solving, planning, decision-making, and learning.

An intelligent agent pursues its goals and executes its actions such that it optimizes some given performance measure. An intelligent agent operates flexibly and rationally in a variety of environmental circumstances, given the information they have and their perceptual and effectual capabilities [1]. A rational or intentional agent is an agent that has explicit goals motivating its action [2]. A rational agent is an intelligent agent.

A multi-agent system is characterized by being comprised of the following elements [2]:

1. An environment at which the agents interact.
2. A set of passive objects situated within the environment, which agents can perceive, create, destroy and modify.
3. A number of agents, representing a specific subset of environment objects, representing system's active entities.
4. A number of relations that link objects and agents to each other.
5. A number of operations that enables agents to perceive, produce, consume, transform and manipulate environment objects.
6. Laws of the universe.

1.2 Communication

Communication forms the basis for interaction and social organization. Communication is a threefold problem involving knowledge of [3]:

(i) **interaction protocol**[1]: conceived as a high-level strategy followed by the agent, governing its interaction with other agents

(ii) **communication language**: the medium through which an agent communicates attitudes regarding the content of the exchange

(iii) **transport protocol:** communication's transport mechanism, such as TCP, SMTP, and HTTP

1.2.1 Communication Levels

There are three levels at which communication protocol are specified [4]:

[1] This definition differs from that adopted by the researcher as described in subsection 1.2.4

- The lowest level specifies the method of interconnection;
- The middle level specifies the syntax of transferred information;
- The top level specifies the information semantics. The semantics refers not only to the substance of a message, but also to the type of the message.

Communication protocols are either binary or n-ary. In a binary protocol, there is a single sender and a single receiver, whereas in an n-ary protocol, there is a single sender and multiple receivers [4].

1.2.2 Speech Acts

The speech acts theory views natural human language as actions. For example, a request, a suggestion, a commitment, or reply is viewed as action. A speech act has the following aspects [4]:

1. **A locution:** defining the physical utterance by the speaker
2. **An illocution:** defining the intended meaning of the utterance by the speaker
3. **A perlocution:** defining the action that results from the locution.

Speech acts can be classified to several types although it is not always considered as being totally definitive [2]:

- **Assertive acts:** intended to give information on the world by asserting something.
- **Directive acts:** intended to give directions to the addressee.
- **Permissive acts:** intended to commit the locator to performing certain acts in the future.
- **Expressive acts:** intended to give the addressee indications of the mental state of the locator.
- **Declarative acts:** intended to perform an act by the mere fact of making the utterance.

1.2.3 Knowledge Query and Markup Language (KQML)

Knowledge Query and Markup Language (KQML) was considered as a message format and a message protocol, supporting run-time knowledge sharing among agents. Following is KQML key features [3]:

- KQML communicate an attitude, such as assertion, request, query, about the content of a message.
- Performatives are language primitives that define permissible actions agents can attempt in communicating with each other.
- In a KQML speaking environment, special types of agents, called facilitators, may exists, providing a set of functions of naming, registration, and communication services.

KQML can be viewed as consisting of three layers [3]:

- The content layer: carry the actual message content in the program's own representation language.
- The communication layer: encodes a set of message attributes that describes lower level communication parameters including: the identity of the sender and recipient, a unique identifier associated with the communication, and so on.
- The message layer: represent the core of the KQML language and determines the interaction types that can be used by a KQML-speaking agent.

1.2.4 Conversations

A conversation can be defined as a series of communications among different agents that follows a protocol and with some purpose [5]. A layered approach for modeling complex concurrent conversations based on Petri-nets was defines the following layers of a conversation model [6]:

- **The protocol layer:** the protocol layer is the basic layer of the model representing an interaction. A protocol is defined as a pattern of sequences of expected communicative acts organized into roles. The protocol identifies the roles, the sequences of communicative acts, and the relations between roles. In this definition, a role is defined as an identity of a single sequence of acts executed by a single entity.
- **The conversation layer:** the conversation layer is the middle layer of the model. A conversation is defined as a particular occurrence of a protocol or a

set of protocols forming a continuing sequence of messages exchanged between two or more agents.

- **The policy layer:** the policy layer is the higher level layer, forming a collection of rules and interaction specifications that guide a particular path in a conversation space. A policy describes constraints that apply to a conversation as it evolves.

A conversation follows one trajectory from a space of possible sequences of communicative acts defined by an interaction protocol and guided by a conversation policy, which may be implemented simply by a set of rules, or in more complex cases, through a policy-level interaction protocol [6].

A conversation can be described using finite-state automata as a series of states linked by transitions representing communications exchanged by agents [2]. Finite-state automata is suitable when a conversation can be summarized as a single process. As agents sometimes involve in several conversations simultaneously, and they have to manage these multiple conversations. It is then easier to describe the nature of these interactions by using Petri-nets, which is very frequently used to describe protocols in distributed systems [2].

1.3 Interactions

An interaction occurs between two or more agents brought into a dynamic relationship through a set of reciprocal actions. An interaction situation is defined as an assembly of behaviors resulting from the grouping of agents acting in order to attain their objectives, paying attention to the resources available to them and to their individual skills [2].

1.3.1 Elements of Interaction

The main elements of interaction situations can be classified into three categories [2]:

- **Compatible and Incompatible Goals:** If goals are compatible, then it is a cooperation or indifference situation, and if goals are incompatible, then it is a situation of antagonism or opposition.
- **Relation to Resources:** The quantity of necessarily limited resources leads to conflicts arising because several agents need the same resources at the same

time and in the same place. Conflict situations can be resolved through coordination of actions and through conflict resolution mechanisms.

- **Capacities of Agents in Relation to Tasks:** In some situations, where certain tasks for which the capacities of several agents are necessary, interaction enables agents to account for this transformation. The product in this case is not a simple sum of individual agent capabilities.

1.3.2 Types of Interaction

Elements of interaction discussed below enable the formation of an initial typology of interaction situations providing for all possible cases. The situations are discussed as follows [2]:

- **Independence:** This type of interaction is characterized by agents with compatible goals, sufficient resources, and sufficient skills. Agents carry out actions independently without any effective interaction. This situation poses no problems from the multi-agent point of view.

- **Simple Collaboration:** This type of interaction is characterized by agents with compatible goals, sufficient resources, and insufficient skills. This type of interaction is achieved by the simple addition of skills that require no supplementary coordination of actions is required. Interactions are expressed in the form of the allocation of tasks and the sharing of knowledge.

- **Obstruction:** This type of interaction is characterized by agents with compatible goals, insufficient resources, and sufficient skills. In this type of interaction situations, agents experience conflicts with other agents as they try to accomplish their tasks with no dependency between their actions.

- **Coordinated Collaboration:** This type of interaction is characterized by agents with compatible goals, insufficient resources, and insufficient skills. This type of interaction assumes that agents have to coordinate their actions to procure the synergic advantages of pooled skills. This type of interaction is the most complex type of cooperation situations. It combines task allocation problems with aspects of coordination shaped by limited resources.

- **Pure Individual Competition:** This type of interaction is characterized by agents with incompatible goals, sufficient resources, and sufficient skills. In this type of interaction, agents have to struggle or negotiate in order to achieve their individual goals, access to resources is not what the conflict is about. There is no specific interaction problems linked to this type of situation.

- **Pure Collective Competition:** This type of interaction is characterized by agents with incompatible goals, sufficient resources, and insufficient skills. In such situation, agents have to group themselves together into partnerships or associations to be able to achieve their goals. Group formation is carried out into two successive phases. At the first phase, individuals ally within groups united by ties of coordinated collaboration. The second phase tends to set one group against another without obstruction.
- **Individual Conflict over Resources:** This type of interaction is characterized by incompatible goals, insufficient resources, and sufficient skills, conflicts situations occur because agents do not want to share resources.
- **Collective Conflict over Resources:** This type of interaction is characterized by incompatible goals, insufficient resources, and insufficient sufficient skills. In this type of situation, collective competition is combined with individual conflicts over resources.

1.3.3 The Analysis of Interaction Situations

Situations can be analyzed at different interaction levels. A complex interaction, or macro, situation is composed of elementary, or micro, situations. The relationship between micro and macro situations is of the parts to the hole. A macro situation, although being the product of micro situations, introduces a set of problems that triggers the creation of a number of micro situations in order to produce their solutions, generating problems themselves [2].

1.3.4 Interaction Protocols[2]

An interaction protocol governs a conversation between a team of agents. Interaction protocols have different objectives. For example, when agents have conflicting goals or when agents are self-interested, the protocol's objective is to maximize agent utilization. On the other hand, when agents share similar goals or tries to solve a common problem, as in distributed problem solving (DPS), the protocol objective is

[2] In this context, the term interaction protocol referss to both the protocol of interaction and the conversation policy, refer to subsection 1.2.4 for more explanation.

to maintain globally coherent performance without violating autonomy meaning that there is no explicit global control [4].

1.3.5 Interaction and Multi-Agent Belief Maintenance

A multi-agent truth maintenance system can serve as a detailed example of high-level interaction among agents. A truth-maintenance system (TMS) is designed to ensure the integrity of an agent's knowledge, which should be stable, well-founded, and logically consistent. Depending on how beliefs, justification and data are represented, a stable state of a knowledge base is one in which (i) each datum that has a valid justification is believed, and (ii) each datum that lacks a valid justification is disbelieved. A well-founded knowledge base permits no set of its beliefs to be mutually dependent. A logically consistent knowledge base is one that is stable at the time that consistency is determined and in which no logical contradiction exists. A consistent knowledge base is one in which no datum is both believed and disbelieved (or neither), or in which no datum and its negation are both believed. Other properties of a knowledge base are that it be complete, concise, accurate, and efficient [4].

A single-agent TMS attempts to maintain well-founded stable states of knowledge base by adjusting which data are believed and which are disbelieved. However, it is important for a group of agents to be able to assess and maintain the integrity of communicated information, as well as of their own knowledge. A multi-agent TMS can provide this integrity [4].

We consider a modified justification-based TMS, in which every datum has a set of justifications and an associated status of INTERNAL (believed, because of a valid local justification), EXTERNAL (believed, because another agent asserted it), or OUT (disbelieved). Consider a network of many agents, each with a partially-independent system of beliefs. The agents interact by communicating data, either unsolicited or in response to a query. For well-founded-ness, a communicated datum must be INTERNAL to at least one of the agents that believes it and INTERNAL or EXTERNAL to the rest [4].

The support status of a communicated datum is jointly maintained by several agents. Hence, a single agent is generally not free to change the status on its own accord. It must coordinate with the other agents so that they are consistent on the status of the datum [4].

The multi-agent TMS is invoked by the addition or removal of a justification, and obeys the following principles [4]:

- Belief changes should be resolved with as few agents as possible.
- Belief changes should be resolved by changing as few beliefs as possible.

When a TMS invoked, it does the following three things [4]:

1. Un-labels some data, including the newly justified datum and, presumably, its consequences. This unlabeled data set might be confined to a single agent or it might span several agents. If a communicated datum is unlabeled in some agent, it must be unlabeled in all the agents that share it.
2. Chooses labeling for all the unlabeled shared data, as defined above.
3. Initiates labeling by each of the affected agents with respect to the requirements imposed by the shared data. If any of the affected agents fail to label successfully, it then backtracks. It either chooses different labeling for the shared data (step 2), or un-labels a different set of data (step 1).

1.4 Coordination

Coordination refers to one of the following aspects [4]:

- A state of an agent community where agents' actions fit well with each other or,
- the process of achieving a state of coordination within an agent community

The degree of coordination is the degree to which agents [4]:

- Avoid live lock and deadlock, and
- Maintaining applicable safety conditions.

Defining Coordination of Actions

"In relation to cooperation, Coordination of actions can be defined as the expression of the individual actions accomplished by each of the agents in such a way that the whole ends up being a coherent and high performance operation. Coordination then is a matter of arranging the behaviors of each agent in time and space in such a way that the group action is improved, either through better performances or through a reduction in conflict" [2].

As coordination occurs between agents, the definition of agents, with which coordination is necessary, is often provided by the problem itself. Interdependencies between the actions of a set of agents should be managed. The Detection of the relationships that exits between two actions is of major importance in organizing a set of actions, and is considered a crucial problem for the organization and improvement of the coordination of actions [2].

Agents coordinate their actions for four main reasons [2]:

1. Agents require information and results other agents' supply.
2. Since resources are limited, they have to be shared in such a way as to optimize the actions to be carried, and in the mean time, try avoiding possible conflicts.
3. Coordination of actions enables cost reduction by eliminating pointless actions and avoiding redundant actions.
4. Agents might have separate interdependent objectives that they need to achieve while profiting from goal interdependencies.

1.4.1 Coordination Systems

Coordination systems posses a number of characteristics that can be classified into the following four categories [2]:

Temporal characteristics: includes rapidity, predictability and predictive-ness

- **Rapidity**: relates to the system's capability to react more or less promptly to a foreseen or unforeseen event. The less a system reasons, the faster it is, and the more it is conceived as 'wired up'.
- **Adaptability**: relates to system's capability to take unexpected events into account. The more adaptable the system is, the more it can be implemented in evolutionary contexts.
- **Predictive-ness**: relates to agent's ability to predict the future, with greater or less precision, the state of the world and of the state of other agents in the future. The more predictive a system is, the less it will tend to be rapid and adaptable.

Organizational characteristics: includes organizational structure, mode of communication, freedom of action left to an agent

- **Organizational structure:** can be described along a centralization/distribution axis, the most centralized organizations are generally the most coherent and the simplest to implement. On the other hand, distributed organizations are able to adapt more easily to unexpected modifications in the environment, and, in particular, to possible malfunctions of certain agents.
- **Mode of communication:** characterizes the way in which the agents acquire knowledge of the actions of the other agents. Agents can perceive other agents either directly by identifying their behavior, or indirectly by means of traces and propagation of signals. Modes of communication play a determining role in the manner in which agents interact with each other since coordination must necessarily go through communication.
- **Freedom of action left to an agent:** with regard to the instructions an agent receives from a coordinator or in relation to the coordination protocol, it characterizes the degree of independence in the agent's behavior. The freer agents are to act as they match, the more they can adapt to special circumstances, but also the more they are in a position to create 'traps', or locally blocking situations, from which it is often difficult to emerge.

Quality and efficiency characteristics: includes quality of coordination, avoidance of conflicts and number of agents

- **Quality of the coordination:** it is the main matter of importance in a coordination system.
- **Avoidance of conflicts:** relates to its capacity for avoiding and emerging from possible conflicts.
- **Number of agents:** measured by the number of agents it is capable of coordinating.

Realization characteristics: includes the degree and quantity of data, the degree of mutual representation required and the difficulty of implementation

- **The degree and quantity of data:** concerns the quantity of data agents have to exchange to coordinate their actions. Exchanges can be carried out at two different times, either at the time plans of actions drawn up, or during execution. Coordination is used since it makes it possible to reduce the number of communications necessary, or to avoid information 'overflow'.
- **The degree of mutual representation required:** as agents require information on the world and on the others, more information is needed to update this knowledge, and hence, it is more difficult to make it appropriate to the state of the environment and that of the other agents. On the other hand, the more information an agent have about other agents, the more it is in a position to predict system's evolution and hence, to respond in a manner adapted to the actions of the different agents.
- **The difficulty of implementation:** it is preferred to keep the simplest coordination techniques since because they are easy to design, realize, and maintain.

Characteristics of Generalization: These characteristics indicate to what extent a coordination method is general, by authorizing some heterogeneity in agents or by applying to different domains.

- **The heterogeneity of the approach:** characterized as the ability to define coordination systems able to take account of agents differing in their perception, reasoning and communications capacities, and therefore, in their capacity of interconnecting agents not supporting the same coordination mechanisms.
- **The generality of the method:** certain techniques are often better adapted to some domains than others. General coordination methods are adaptable to any domain but, on the other hand, the more general a technique is, the less efficient it usually is.

1.4.2 Coordination Models

In order to produce coordinated systems without bottlenecks, and exhibiting graceful degradation of performance, most DAI research has concentrated on techniques for having both control and data distributed. Distributed control means that individual agents have a degree of autonomy in generating new actions and in deciding which goals to work at next. The disadvantage of this approach is that knowledge of the system's overall state is dispersed throughout the system, leading to a situation where each agent has only a partial and imprecise perspective. Increased uncertainty about each agent's actions exists, causing more difficulty attaining coherent global behavior. In order to ensure that agents can correctly reason about coordination, a number of coordination models were developed, allowing individuals to reason about the way their actions shall contribute to the collective problem solving effort and the way they can benefit from their interactions [7, 4].

1.4.2.1 Modeling Coordination as a Distributed Goal-Search Problem

Several researchers have characterized DAI as a form of distributed goal search [7], where actions of agents in solving goals can be expressed as search through a classical AND/OR goal graph which includes a representation of the dependencies between the goals and the resources needed to solve the primitive goals (leaf nodes of the graph). Indirect dependencies can exist between goals through shared resources interactions [4]. The formulation a multi-agent system in this manner allows clear identification of the activities that requires coordination that includes [7, 4]:

- The definition of a the goal graph, which includes the identification and classification of dependencies
- Assignment of particular graph regions to appropriate agents
- The control of decisions about which areas of the graph to explore
- Graph traversal
- Ensures the reporting of successful traversal

Some of the activities are collaborative, while the others are carried out by an individual agent acting in isolation. Determining the approach for each of the phases is a matter of the multi-agent system design [7, 4].

1.4.2.2 Commitments and Conventions as the Foundations of Coordination

Commitments and conventions are the key agent structures used by an agent. Commitments are regarded as promises to undertake a specified course of action, conventions provides a means of managing commitments in changing circumstances.

Commitments provide a degree of predictability that enables agents to take future activities of other agents into consideration when dealing with inter-agent dependencies, global constraints, or resource utilization conflicts. As situations change, agents must be able to evaluate whether existing commitments are still valid. Conventions help constrain the conditions under which commitments should be reassessed, and specifies actions associated that should then be undertaken which includes, retaining, rectifying or abandoning commitments. Conventions do not specify how an agent should behave towards other in case that its commitments have been altered or modified [7, 4].

1.4.3 Forms of Coordination

Coordination of actions has several forms classified as follows [2]:

- **Coordination by synchronization:** this form of coordination method is the most elementary and low-level form of coordination where all coordination of actions should precisely describe the sequencing of actions, leading to a necessary synchronization in their execution. This form is required in case that there is a need to manage the concurrence of several actions and to verify that the results of the operations are coherent.
- **Coordination by planning:** this is the most traditional form of coordination in AI, based on breaking down actions into phases. Actions required to achieve a goal and produce a plan are considered in the first phase. A plan is selected for execution in the second phase. Plans might require change plans they are being carried out due to environment changes, requiring the existence of dynamic re-planning options. In addition, in multi-agent systems, as different plans drawn by the agents may lead to conflicts of objectives or conflicts over access to resources, plans must be coordinated in order to solve these conflicts and so achieve the goals of different agents. Although planning techniques provide, in general, for high quality coordination, they prove to be incapable of taking account of unforeseen or extremely complex situations.
- **Reactive coordination:** a more recent coordination technique that considers that coordination mechanisms based on reactive agents are often easier to implement than to those based on planning.

- **Coordination by regulation:** This method is practical in systems requiring limited coordination. The basic concept is to set the rules of behaviors that aim to eliminate possible conflicts.

1.4.4 Coordination by Regulation (Based on Market Mechanisms)

Market-based coordination mechanisms for coordination are needed when there are a large or unknown number of agents [1]. Market-based computational economics are effective for coordinating the activities of many agents with minimal direct communication among them. The research challenge is to build computational economics to solve specific problems of distributed resource allocation [4].

In a market-based environment, everything of interest to an agent is described by current prices – the preferences or abilities of others are irrelevant except in as far as this as they (automatically) affect the prices. In such an environment, there are two types of agents, consumers, who changes goods, and producers, who transform some goods into other goods. Agents bid for goods at various prices, but all exchanges occur at current market prices. All agents bid so as to maximize either their profits or their utility [4].

To cast a problem in terms of a computational market, one needs to specify the following entities [4]:

- the goods being traded
- the consumer agents that are trading the goods
- the producer agents, with their technologies for transforming some goods into others
- the bidding and trading behaviors of the agents

Since markets for goods are interconnected, the price of one good will affect the supply and demand of others. The market will reach a competitive equilibrium such that (i) consumers bid to maximize their utility, subject to their budget constraints, (ii) producers bid to maximize their profits, subject to their technological capability, and (iii) net demand is zero for all goods [4].

The important property is that equilibrium corresponds – in some sense optimally – to an allocation of resources and dictates the activities and consumptions of the agents.

In general, equilibrium need not exist or be unique, but under certain conditions, such as when the effect of an individual on the market is assumed to be negligible, they can be guaranteed to exist uniquely [4].

In an open market, agents are free to choose their strategy, and they do not have to behave rationally. Economic rationality assumes that the agent's preferences are given long with knowledge of the effects of the agent's actions. From these, the rational action for an agent is the one that maximizes its preferences [4].

Economic rationality has the charm of being a simple, "least common denominator" approach – if you can reduce everything to money, you can talk about maximizing it. But to apply it well requires a careful selection of the target problem [4].

One of the oldest applications of economic rationality is in decision-theoretic planning, which models the costs and effects of actions quantitatively and probabilistically. For many applications, where the probabilities can be estimated reliably, this leads to highly effective plans of action [4].

The need to maximize preferences essentially requires that there be a scalar representation for all the true preferences of an agent. In other words, all of the preferences must be reduced to a single scalar that can be compared effectively with other scalars. This is often difficult unless one can carefully circumscribe the concepts under a veneer of rationality. For example, if we would like an agent to be governed by its past commitments, not just the most attractive choice at the present time, they we can develop a utility function that gives additional weight to past commitments. This approach may work in principle, but in practice, it only serves to hide the structure of commitments in the utility function that one chooses [4].

1.5 Cooperation

We adopt the following definition for cooperation:

"Cooperation is defined as coordination among non-antagonistic agents where participants succeed or fail together…" [4].

If one of the following two conditions is verified, then a group of agents are in a cooperation situation [2]:

- Adding a new agent could result in an increase in performance levels of the group or,

- agent actions serve to avoid or to solve potential or actual conflicts.

There are several types of cooperation methods that can be classified into the following six categories [2]:

1. **Grouping and multiplication:** the grouping method allows a group of agents to behave as if they were side by side physically, forming a homogeneous in space or communication network, by arranging agents to draw physically closer together. Multiplication is the quantitative increase in the number of individuals in a given system.
2. **Communication:** by allowing agents to benefits from the information and know-how that other agents possess, communication expands the perspective capacities of agents. Communications are indispensable in cooperation.
3. **Specialization:** specialization is defined as the process through which agents become more and more adapted to their tasks.
4. **Sharing tasks and resources:** Mechanisms of supply and demand are the means of tasks distribution in cognitive systems. Such a mechanism is either in cognitive systems, having a coordinating agent centralizing the supply and demand and then distributing them to the best advantage, or distributed, where all agents can be offering supplies and making demands at the same time without any centralizing agent.
5. **Coordination of actions:** given an assembly of agents pursuing their own goals, coordination tasks are indispensable.
6. **Conflict resolution by arbitration and negotiation:** arbitration and negotiation are two means with an overall effect of limiting conflicts and preserving individuals and societies through resolving conflicts, preventing occurrence of open struggle due to disagreements between individuals, and maintaining system's performance levels. Arbitration enables the definition of rules of behavior that acts as constraints on an assembly of agents.

1.5.1 Cooperation Protocols

Many cooperation protocols share a basic strategy based on decomposition of tasks, then distributing them. This approach enables the reduction of task complexity since subtasks require less capable agents and fewer resources. The system must decide among alternative decompositions if available, moreover, the decomposition process must take into consideration agent capabilities and resources available to them [4].

The task decomposition can be done using several approaches [4]:

- by the system designer at the system design time and resulting in decomposition being programmed during system implementation
- by several agents using hierarchical planning
- inherent in the problem representation as in an AND-OR graph
- spatially based on the layout of information sources or decision points
- functionally, according to agents' available expertise.

As tasks being decomposed, they can be distributed according to several criteria [4]:

- prevent overloading of critical resources
- assigning tasks based on matching agent capabilities
- a controller agent with wider view assigns tasks to other agents
- the assignment of overlapping responsibilities to agents in order to achieve coherence
- agents in a spatial or semantic proximity are assigned highly interdependent tasks
- if necessary, urgent tasks might be reassigned

The following mechanisms are commonly used to distribute tasks [4]:

- **Market mechanisms:** using an approach analogous to pricing commodities, tasks are matched to agents by generalized agreement or mutual selection
- **Contract net:** agents uses announce, bid, and award cycles, see 1.5.2
- **Multi-agent planning:** the responsibility of task assignment is given to planning agents
- **Organizational structure:** in a multi-agent system organization, agents have fixed responsibilities for particular tasks

1.5.2 The Contract Net Protocol

The contract net protocol is an interaction protocol that is widely applied for cooperative problem solving. A manager agent, with a desire to solve a task, interacts with a set of potential contractor agents that might be able to solve tasks. The contracting process can be conceived as follows [4]:

1. Manager announce to contractors a task that needs to be performed
2. Contractors receive task announcements
3. Contractors evaluate their self capabilities to respond
4. Contractors respond by declining or bidding
5. Manager Receive and evaluate bids from potential contractors
6. Manager award a contract to a suitable contractor
7. Contractors perform the task if my bid is accepted
8. Contractors report my results
9. Manager receive and synthesize results

Agent roles are not specified in advance but rather, any agent can act as a manager by making task announcements, and any agent can act as a contractor by responding to task announcements. This approach enables flexibility allowing for further task decomposition as a contractor for a specific task may act as a manager by soliciting the help of other agents in solving parts of that task. The resulting manager-contractor links form a control hierarchy for task sharing and result synthesis [4].

1.6 Negotiation

Negotiation is defined as interaction between agents based on communication for the purpose of coming to an agreement, a process by which a joint decision is reached by two or more agents, each trying to reach an individual goal or objective. Negotiation can also be perceived as coordination among competitive or simply self-interested agents, or as a distributed communication-based search through a space of possible solutions. Negotiation is much related to distributed conflict resolution and decision-making, requiring agents to use a common language for making proposals which is then commented (refined, criticized, or refuted) by other agents which start by communicating their possibly conflicting positions, and then trying to move towards agreement by making compromises or searching for alternatives [1]. Negotiation supports cooperation and coordination between agents [8].

Negotiation mechanisms have two categories, environment-centered or agent-centered. Environment-centered mechanisms should ideally have the following attributes [4]:

- **Efficiency:** an agent should not waste resources in coming to an agreement.
- **Stability:** an agent should not have an incentive to deviate from agreed upon strategies.
- **Simplicity:** the negotiation mechanism should entail low computational and bandwidth demands on the agents.
- **Distribution:** the negotiation mechanism should not require a central decision maker.
- **Symmetry:** the negotiation mechanism should not be biased against any agent for arbitrary or inappropriate reasons.

Most agent-centered negotiation mechanisms are developed for specific problems, agent-centered negotiation mechanisms can be categorized into two general approaches. The first approach is based on speech acts classifiers together with possible-worlds semantics that formalize negotiation protocols and their components, clarifying conditions of satisfaction for different kinds of messages. The second approach is based on an assumption that agents are economically rational. Using this approach, the set of agents must be small, with common problem abstractions, with a goal of reaching a common solution [4].

In a closed environment, participating agents are controlled in advance and hence, the protocol can be assumed to be known and fixed. On the other hand, such assumption is invalid in an open environment, since each time agents interact, they will have to recognize, construct, or select an appropriate protocol for that specific occasion [9].

1.7 Multi-Agent Organization

Agents, either human or artificial, need social and organizational intelligence in order to navigate through an organizational world. This organizational intelligence cover many dimensions, including communication capabilities, knowledge about which agent knows what, knowledge about environment rules, procedures, and organizational culture [10].

We adopt the following definition for organization [2]:

"...an arrangement of relationships between components or individuals which produce a unit, or system, endowed with qualities not apprehended at the level of the components or individuals. The organization links, in an inter-relational manner, diverse elements or events or individuals, which thenceforth become the components of a whole. It ensures a relatively high degree of interdependence and reliability, thus providing the system with the possibility of lasting for a certain length of time, despite chance disruptions"

The term organization defines both the process of building up a structure and the actual result of this process. This duality of meaning points up the dynamic aspect of all organizations, which are necessarily dynamic, and it is always in the process of reorganizing the assembly of entities and the links uniting them [2].

In multi-agent systems, numerous interrelationships exist among the agents, such as the delegation of tasks, transfer of data, obligations, synchronization of actions, and so on. These interrelationships are possible only within an organization that needs them in order to exist [2].

Concepts of an agent and that of a multi-agent system can be integrated, since from the designer point of view, an agent is not just an individual, but also an assembly of components. Similarly, a multi-agent system can be considered as a composition of agents and in the same time as a unit, with interactions with another technical device that can be studied. By using the concept of levels of organization it is possible to consider the embedding of one level into another, considering that an organization is an aggregation of elements of a lower level and a component in organizations of a higher level [2].

The intelligence of agents within an organization affects its ability to act. However, organizations, and multi-agent systems in general, often demonstrates different intelligence and capabilities from those of the agents within them [10].

Generally, organizations are characterized as [10]:

- problem solving technologies of large-scale
- encompassing of multiple human and/or artificial agents
- engaged in one or more tasks
- directed by goals that can have the property of being changeable, not articulated, or shared between organizational members.
- affected by and affecting environment
- possess knowledge, culture, memories, history, and capabilities that are distinct from any single agent
- possess legal standing distinct from that of individual agents.

1.8 Distributed Problem Solving

In this section, we introduce the concept of distributed problem solving. We first start by introducing distributed problem solving and then, we discuss four techniques used namely, task sharing, result sharing, distributed planning, and distributed planning and execution.

1.8.1 Introduction

Distributed problem solving (DPS) is the name applied to a subfield of distributed artificial intelligence (DAI) in which the emphasis is on getting agents to work together well to solve problems that require collective effort. Distributed problem solving presumes the existence of problems that need to be solved and expectations about what constitute solutions [11].

Distributed problem solving can be classified into two categories [2]:

- The first category assumes that the total expertise is distributed among all agents, each having only restricted skills in relation to the complete problem.
- The second category assumes that the problem itself is distributed, where agents having similar skills.

In addition to these two categories, another class of problem exists where agents are used in interaction to solve classical problems [2].

Solving distributed problems well demands (i) group coherence, where agents need to want to work together, and (ii) competence, where agents need to know how to work together well. In DPS, the presence of a fair degree of coherence is assumed in the sense that the agents have been designed to work together; or that the payoffs to self-interested agents are only accrued through collective efforts; or that the social engineering has introduced disincentives for agent individualism [11].

Sometimes the problem the agents are solving is to construct a plan, and often even if the agents are solving other kinds of problems, they also have to solve planning problems as well. That is, how the agents should plan to work together – decompose problems into sub-problems, allocate these sub-problems, exchange sub-problem solutions, and synthesize overall solution – is itself a problem the agents need to solve. Distributed planning is thus tightly intertwined with distributed problem solving, being both a problem in itself and a means to solving a problem [11].

Next we shall describe four strategies for DPS, task sharing, result sharing, distributed planning, and distributed planning and execution.

1.8.2 Task Sharing

Task sharing is the first strategy for (DPS). The idea behind this strategy is that when agent has many tasks to do, it should enlist the help of agents with few or no tasks [11].

In cognitive systems, tasks are distributed by means of supply and demand mechanisms. In centralized distribution mechanisms, a coordinating agent centralizes the supply and demand and then distributes tasks to the best advantage. In distributed approaches, all agents can be offering supplies and making demands at the same time without any centralizing organ. Given the distributed approach, there are two contrasting techniques. The first technique is based on acquaintance networks, where a mutual representation of agent abilities exists. The second technique is based on market concepts, where an example protocol is the contract net protocol [2].

The main steps in task sharing are [11]:

1. **Task Decomposition:** generate the set of tasks to potentially be passed to others. This could generally involve decomposing large tasks into subtasks that could be tackled by different agents.

2. **Task Allocation:** assign subtasks to appropriate agents.

3. **Task Accomplishment:** the appropriate agents each accomplish their subtasks, which could include further decomposition and sub-subtask assignment, recursively to the point that an agent can accomplish the task it is handed alone.

4. **Result Synthesis:** when an agent accomplishes its subtasks, it passes the result to the appropriate agent, which is usually the original agent, since it knows the decomposition decisions and thus is most likely to know how to compose the results into an overall solution.

1.8.3 Result Sharing

By sharing results, problem solvers can improve group performance in combinations of the following ways [11]:

1. **Confidence:** having independently derived results for the same task enables corroborating each other, leading to a collective result that has a higher confidence of being correct.

2. **Completeness:** each agent formulates results for whichever subtasks assigned to it, which can be combined altogether in order to cover a more complete portion of the overall task.

3. **Precision:** an agent needs to know more about the solutions that other agents have formulated in order to refine its own solution.

4. **Timeliness:** solving subtasks in parallel can yield an overall solution faster.

1.8.4 Distributed Planning

Distributed planning can be thought of simply as a specialization of distributed problem solving, where the problem being solved is to design a plan. However, because of the particular features of planning problems, it is generally useful to consider techniques that are particularly suited to planning [11]. Next, we introduce four categories for distributed planning techniques.

- **Centralized Planning for Distributed Plans:** Centralized planning for multiple of agents assumes the existence of a single planner agent acting as the plan coordinator [2]. A centralized coordinator agent with such a plan can break into separate threads that it passes to agents that can execute them. Given the assumption that the plan is followed suitably, and assuming the correctness of knowledge and predictability of the world, agents operating in parallel shall achieve a state of the world consistent with the goals of the plan [11].

- **Distributed Planning for Centralized Plans:** This form of distributed planning assumes that formulating a complex plan can be treated as a complex problem that requires the generation of a solution. Such a problem can be modeled as collaboration among a variety of *cooperative planning* specialists, where the planning process is distributed among them, each contributing pieces to the plan, until a plan is created [11].
- **Distributed Planning for Distributed Plans:** This form of distributed planning is conceived as the most challenging version where both the planning process and its results are intended to be distributed. To have a multi-agent plan represented in its entirety anywhere in the system might be unnecessary. Distributed pieces of the plan should be compatible, which at a minimum means that the agents should not conflict with each other when executing plans, and preferably should help each other achieve their plans when it would be rational to do so [11].

1.8.5 Distributed Planning and Execution

Following is the strategies for combining coordination, planning, and execution [11]:

- **Post-Planning Coordination:** The distributed planning approach, characterized by having the planning, coordination, then execution processes occurring in sequences, is a reasonable approach given that agents individually build plans that are likely to be able to be coordinated, and that the coordinated result is likely to be executed successfully. On the other hand, if during execution, one or more plans for agents fail to progress as expected, the coordinated plan set might fail as a whole [11]. One approach to remedy such cases is through contingency planning, having agent formulating alternative plans or branches in order to respond to possible contingencies that would arise during execution. These larger plans can then be merged and then coordinated through a more complicated coordination process. A second mean for dealing with dynamics is through monitoring and re-planning, where the plan-coordinate-execute process is repeated. On the other hand, if this process occurs frequently, it would cause significant overheads. In order to make the effort manageable, a strategy might be to *repair* previous plans, or to access a library of reusable plans. If a plan deviation can be addressed locally, it can save a huge overhead since no coordination is required. For example, by coordinating plans at an abstract level, an agent can re-plan details at execution time without requiring coordination with others as plan revision fits within the coordinated abstract plan [11].

- **Pre-Planning Coordination:** Pre-planning coordination enables agents to coordinate their activities through following a *social law*, a set of social rules acting as a body of constraints that prohibits against particular choices of actions in particular contexts. Social rules entail the existence of highly developed cognitive agents with some form of *free will* [2]. Social laws can be derived by working from undesirable states of the world backwards to find an arrangement of actions leading to those states, and then imposing restrictions on actions so that the combinations cannot arise. Finding states without handcuffing agents from achieving states that are acceptable and desirable poses a challenge to agent developers. On the other hand, in domains were conflict avoidance is not an issue, an agent might prefer to take actions that benefit the society as a hole, even if not these actions are not directly relevant to the agent's goals [11].
- **Interleaved Planning, Coordination, and Execution:** In this approach, planning and coordination are interleaved with each other, and often with execution as well through the use of a hierarchical protocol [11]. The Partial Global Planning (PGP) is a hierarchical protocol that provides a framework for distributed planning for multiple agents. In this model, each agent has a knowledge base that is distributed over three levels. The first level represents local plans, which organizes agent's future activities with all the required details such as short and medium term goals, costs, durations, and so on. At the second level are plan nodes, a node corresponds to an agent, summarizing local plans with unnecessary details being eliminated. At the third level are the partial global plans or PGPs, containing general information on the

developments in part of a global activity. A PGP is a data structure that agents use to exchange information about their objectives and plans, containing information about the set of plans that is responsible for merging, pursued objectives, agents current activities, expected costs and results, the way agents should follow in order to interact. Agents exchange PGPs with others and starting from a model of itself and the others, it attempts to identify agents having goals that form a part of a group objective, called a Partial Global Goal (PGG) and then, combine associated PGPs into a more general PGP to meet this objective [2].

Chapter 2
Multi-Agent Collaboration (MAC)

In order to identify the requirements of a collaboration framework, we need to have a deep understanding of multi-agent collaboration. In this chapter, we discuss the characteristics of multi-agent collaboration. In section 2.1, we start by defining the concept of multi-agent collaboration, and next in section 2.2, we discuss the need for generic models of teamwork. In section 2.3, we introduce three formal theories of teamwork, and finally, we discuss two general models of teamwork in section 2.4.

2.1 Defining MAC

In discussing multi-agent collaboration, we adopt the following definition:

"...forms of high-level cooperation that requires the (development of) mutual understanding and a shared view of the task being solved by several interacting entities..." [1].

Collaboration occur within a team of agents cooperating to achieve some collective goal. As a team of cooperating agents, participating agents succeed or fail together. Sharing a mental state within a team of agents enables reasoning about their beliefs, goals, actions, commitments, and intentions and hence, reason about the success or failure of teamwork.

2.2 The Need for Generic Models of Teamwork

It has been argued by [12] that in dynamic complex domains coordination is not enough to reach teamwork. In the Attack domain [12], carefully preplanned coordination was sufficient to demonstrate desired behavior for small teams of two pilots and a small number of vehicles, significant number of unanticipated agent interactions were surfaced as the number of agents and vehicles increased, their behaviors were enriched, and domain experts began to specify complex missions. The first approach used to overcome this problem was through the addition of domain-specific coordination plans. However, due to several difficulties, coordination plans had to be added on a case by case basis. In addition, as teamwork scenarios become more complex and failures continues to occur. The second approach was based on providing agents with a general model of teamwork that enables agents themselves to reason about coordination and communication as well as anticipate and avoid or recover from teamwork failures [12].

We can classify the causes of team breakdowns into two categories: the lack of mutual understanding of the situation due to communication problems and the problem of maintaining shared mental state causing incorrect assessment of situations by team agents, and the lack of a coordination mechanism. Some of the possible breakdowns in the Attack domain where outlined in [12].

2.3 Multi-Agent Collaboration Theories

Next, we discuss a number of formal theories of teamwork. In 2.3.1, we discuss the joint intentions theory. We discuss the shared plans theory in 2.3.2 and finally, we discuss the cooperative problem solving theory in 2.3.3.

2.3.1 The Theory of Joint Intention

The Joint Intentions theory represents one of the first attempts to establish a formal theory of multi-agent collaboration, and due to its clarity and expression, is probably the best known of the rational agent theories [13]. The theory defines logic of rational action which has generated a lot of interest in the world of multi-agent universes, and which led to a lot of research based on the formalism and the foundations of this theory. The theory furnishes a uniform and practical notation to link concepts such as intentions, beliefs and action putting forward a relatively integrated concept of rational action [2]. The theory is not intended to be directly implemented, but to be used as a specification of agent design [12].

The basic argument is that a joint activity is one that is performed by individuals sharing certain specific mental properties which affect and are affected by properties of the participants [14]. In addition, the theory argues the benefit of teamwork by showing that in return for the overhead involved in participating in a joint activity, an agent expects to be able to share the load in achieving a goal in a way that is robust against certain possible failures and misunderstandings [14].

There are two types of intention: the intention to act in the future and the intention to do something now. The two types of intention do not coincide completely, referring back to two different forms of intentionality. The first intentional type assumes a planning mechanism, whereas the second one refers only to the action execution mechanism [2]. The following are the characteristics necessary for a theory of intentional act [2]:

(i) intentions are at the origin of problems that agents have to solve,

(ii) intentions have to be coherent with each other,

(iii) agents should follow up the results of their actions as they remain ready to re-plan their actions to fulfill their intentions, and

(iv) there should be a distinction between the results intentionally wiled by an agent and those which are merely involuntary consequences of the action. It is also assumed that an intention persists as long as the preceding conditions are verified [2].

The theory of joint intentions adopts a methodological concern for treating the future-directed properties of intention as primary, and the intention-in-action properties as secondary. The theory is concerned with how adopting an intention constrains the agents' adoption of other mental states. The theory proposes a set of definitions for individual commitment, individual intention, weak achievement goal, and joint persistent goal [14].

The formal account of individual and joint commitments and intentions is given in terms of beliefs, mutual beliefs, goals, and events. At the very lowest level, the formal account is formulated in a modal quantificational language with a possible-world semantics built out of the following primitive elements described in general term [14]:

- **Events**: It is assumed that possible worlds are temporally extended into the past and future, and that each such world consists of an infinite sequence of primitive events, each of which is of a type and can have an agent.

- **Belief**: Belief is what an agent is sure of, after competing opinions and wishful thinking are eliminated. This is formalized in terms of an accessibility relation over possible worlds in the usual way: the accessible worlds are those the agent has ruled capable of being the actual one. Beliefs are the propositions that are true in all these worlds. Although beliefs will normally change over time, it is assumed that agents correctly remember what their past beliefs were.

- **Goal**: The notion of We have formalized the notion of goal also as accessibility over possible worlds, where the accessible worlds have become those that the agent has selected as most desirable. Goals are the propositions that are true in all these worlds. As with belief, we presume that conflicts among choices and beliefs have been resolved. Thus, we assume that these chosen worlds are a subset of the belief-accessible ones, meaning that anything believed to be currently true must be chosen, since the agent must rationally accept what cannot be changed. However, one can have a belief that something is false now and a goal that it be true later, which is what we call an achievement goal. Finally, we assume agents always know what their goals are.

- **Mutual belief**: The concept of mutual belief among members of a group is taken to be the usual in finite conjunction of beliefs about other agents' beliefs about other agents' beliefs (and so on to any depth) about some proposition. Analogous to the individual case, we assume that groups of agents correctly remember what their past mutual beliefs were.

The theory provides the following definitions:

Individual Commitment

Defined as a notion of individual commitment, an agent has a joint persistent goal, a, relative to q, to achieve p iff [14]:

1. The agent believes that p is currently false;
2. The agent wants p to be true eventually;
3. It is true, and the agent knows it, that (2) will continue to hold until the agent comes to believe either that p is true, or that it will never be true or that q is false.

Some important points to observe about individual commitments are as follows [14]:

- Once adopted, an agent cannot drop them freely; the agent must keep the goal at least until certain conditions arise
- Other goals and commitments need to be consistent with them;
- Agents will try again to achieve them should initial attempts fail.
- Clause 3 states that the agent will keep the goal, subject to the aforementioned conditions, in the face of errors and uncertainties that may arise from the time of adoption of the persistent goal to that of discharge.
- Condition q is an irrelevance or escape clause that the persistence goal is relative to. If the agent comes to believe it is false, it can drop the goal.

- It is important to notice that an agent can be committed to another agent's acting.

Individual intention

An intention is defined to be a commitment to act in a certain mental state. An agent intends relative to some condition to do an action just in case it has a persistent goal relative to that condition, of having done the action and, moreover, having done it, believing throughout that it is doing it [14].

Intentions inherit all the properties of commitments, for example: tracking, consistency with beliefs and other goals. In addition, because the agent knows it is executing the action, intention inherits properties that emerge from the interaction of belief and action. An intention would typically arise within a subgoal-supergoal chain as a decision to do an action to achieve some effect [14].

Weak achievement goal

An agent has a weak achievement goal (WAG) relative to escape condition q and with respect to a team to bring about p if either of these conditions holds: [14]

- The agent has a normal achievement goal to bring about p. The agent does not yet believe that p is true and has p eventually being true as a goal.
- The agent believes that p is true, will never be true, or is irrelevant (q is false), but has a goal to have the status of p mutually believed by all the team members.

Joint persistent goal

A team of agents have a joint persistent goal (JPG) relative to q to achieve p just in case: [14]

1. They mutually believe that p is currently false
2. They mutually know they all want p to eventually be true
3. It is true and mutual knowledge that until they come to mutually believe about the status of p (true, will never be true, or q is false), they will continue to mutually believe that they each have p as a weak achievement goal (WAG) relative to q and with respect to the team.

If a team is jointly committed to achieving p, they mutually believed initially that they each have p as an achievement goal. As time passes, a team member cannot conclude that each other member still have p as an achievement goal, rather, it can only conclude that they have it as a weak achievement goal (WAG). The reason is that each member allows that any other member may have discovered privately that the goal is achieved, unachievable or irrelevant, and be in the process of making that known to the team as a whole. If at some point it is no longer mutually believed that everyone still has the normal achievement goal, the condition for a joint persistent goal (JPG) will no longer holds, even though a mutual belief in a weak achievement goal will continue to persist [14].

Joint intention

Joint intention is defined as a joint commitment to agents having done a collective action, with agents of primitive events as team members in question, and with the team acting in a joint mental state [14]. A team of agents jointly intends, relative to some escape condition q, to do an action if [14]:

1. The members have a joint persistent goal (JPG) relative to that condition of their having done the action
2. Having done it mutually, believing throughout that they were doing it.

Some drawbacks

The following are some drawbacks of the theory as defined in [2]:

"this type of theory has the advantage that it clearly theory outlines some of the key concepts of intention, it has a tendency to be extremely simplistic with regard to the problems of action; in particular it does not take account of revisions of actions or integrate the mechanisms for developing these intentions the conflicts which can arise between motivations of various kinds and constraints. In addition, the theory assumes that a 1:1 relationship exists between the goals and the sequence of actions that can achieve the goals…"

2.3.2 The Shared Plans Theory

The SharedPlans collaboration model was developed to account for several deficiencies noted in Pollack's mental state of plans [13]:

1. An action being performed by two or more agents could be decomposed into actions to be performed by each individual agent.
2. An agent was not required to establish a commitment towards the success of a collaborative partner's actions. If such commitment does not exist, an agent

planning shall not be constrained in regard to another agent's needs or activities.

3. Agents undertaking joint activities frequently do not have an entire roadmap for their activities throughout the course of collaboration. In addition, if they start with a fully-specified plan, an uncertain and dynamic world could obviate some fraction of that plan and hence, the need to maintain a collaboration in the face of incomplete plans.

In contrast with the theory of joint intentions, the SharedPlans theory defines the concept of a SharedPlans (SP). A SP is either a full shared plan (FSP) or a partial shared plan (PSP). A full shared plan to do an action represents a situation where every aspect of the joint activity is fully determined. Aspects of joint activity include mutual beliefs and agreement in the complete recipe to do action. A recipe is defined as a specification of a set of actions which when executed under specified constraints, constitutes performance of action [12].

A SP is not based on joint mental attitudes but relies on a novel intentional attitude, intending that. An individual agent's intention that attitude is directed towards its collaborator's actions or towards a group's joint action. The Intention that attitude is defined via a set of axioms that guide an individual to take actions, including communicative actions enabling its teammates, sub-team or team to perform assigned tasks [12].

The SharedPlans theory aspires to describe the entire web of a team's intentions and beliefs when engaged in teamwork. A PSP is a snapshot of the team's mental state in a particular situation in their teamwork, and further communication and planning is often used to fulfill the conditions of a FSP [12].

2.3.3 The Theory of Cooperative Problem Solving

The theory of cooperative problem solving process presents a model of cooperative problem solving (CPS) represents the best recognized and most influential formal theory of CPS [15]. The theory presents a model of cooperative problem solving (CPS) that describes the process as consisting of four stages. The proposed approach characterizes agents' mental states leading them to solicit, and take part in, cooperative action. The model is formalized by expressing it as a theory in a quantified multi-modal logic. The theory is primarily intended as a specification for future cooperative systems. The model is both abstract and idealized in the sense that there are cases that it does not consider, and some assumptions have been made that are either too strong or too weak [16].

Given the many aspects of CPS investigated by researchers from distributed artificial intelligence, economics, philosophy, organization science, and the social sciences. The models can be divided into two broad categories [16]:

- Implementation-oriented models for (i) realizing cooperative systems, (ii) managing cooperative activities, and (iii) achieving coordination in cooperative systems at runtime. Implementation-oriented models are useful in that they help to identify the various steps of the CPS process.
- Formal theories of cooperation and related issues including economic and game-theoretic models of cooperation and negotiation, formal models of communication based on speech act theory, and models which typically use a multi-modal logic to describe the mental state of agents engaged in social activities.

2.3.3.1 Desiderata for an Adequate Theory of the Cooperative Problem Solving Process

A desideratum for an adequate theory of the cooperative problem solving process is found in [16]:

- *Agents are autonomous*: Agents are autonomous problem solvers and hence, they will take part in cooperative activities only if they choose to do so. A theory that simply required agents to cooperate whenever they were asked to would not be adequate, because it would fail to capture a significant proportion of real-world examples of cooperative activity.
- *Cooperation can fail*: A corollary of the fact that agents are autonomous is that cooperation may fail. Failure might occur even when initial cooperation is established due to many different reasons. Alternatively, successful completion of cooperation may be impossible due to events that are beyond the control of the team. A theory of cooperation is said to be adequate if it (i) recognize that such failure is possible, (ii) identify the key points at which failure may occur, and (iii) characterize the behavior of a rational agent in such situations.

- *Communication is essential*: communication is so fundamental to the process of cooperation. An adequate theory should describe when and where communication should take place. That is, it should predict communication.
- *Communicative acts are characterized by their effect*: An adequate theory of cooperation should not set down the means through which communication actually takes place.
- *Agents initiate social processes*: An adequate theory of cooperation should account circumstances under which agents will begin to initiate cooperation and circumstances under which agents will initiate the social processes required to instantiate and complete cooperative actions.
- *Agents will be mutually supportive*: Cooperating agents will execute their part of the team's action, and will do what they can to ensure that the remainder of the team does likewise. An adequate theory of cooperation must describe the types of mutual support, at what time it should occur, and the form such support should take.
- *Agents are reactive*: Agents must recognize that any realistic environment is highly dynamic and hence, they must respond accordingly to any changes that affect their plans. An adequate theory of cooperation must therefore recognize this reactive aspect of rational behavior, and characterize the behavior of the agents in such circumstances.

The model of cooperative activity is based on the internal perspective, where the agent's internal state is used as the basis for evaluation. Reasons to use this perspective are as follows [16]:

- it provides a high-level specification tool for the cooperating agent designer since it identifies the agent's key data structures, the relationships between these structures, and places some constraints on the values which the structures can take. The approach's prescriptive nature contrasts with external models, mainly concerned with developing theories about agents, rather than on models which agents might use.
- With the external perspective, it is sometimes difficult to distinguish between actions that are coordinated non-cooperative actions, and actions that are truly cooperative as in the later case, participating agents have a collective goal.

2.3.3.2 Structures Controlling an Agent's Cooperative Problem Solving Activities

The model characterizes structures controlling an agent's cooperative problem solving activities into two categories: (i) structures related to individual behavior, and (ii) structures responsible for guiding social behavior [16].

The CPS model requires two types of individual and societal features of mental states to exist. The individual feature suggests that joint action can be reduced solely to individual mental states; whereas the societal feature suggests that individual behavior is equivalent to social behavior in which the groups have precisely one element. Although group constructs including teams, joint goals, joint commitments …etc., are a natural tool for describing social activity since. On the other hand, it is the individuals who ultimately have the ability to act. As a result, there must be a clear mapping to the individual mental states of the participating agents. The CPS model therefore defines social attitudes in terms of individual attitudes. Individual beliefs and goals are taken to be primitive, while other constructs, including those which characterize collective mental states are defined in term of them [16].

The key mental states that control agent behavior in this model are intentions, defining asocial behavior, and joint intentions, controlling social behavior. Intentions provide both the stability and predictability that is necessary for social interaction, and the flexibility and reactivity that are necessary to cope with a changing environment. The model defines a commitment, a pledge or a promise, and defines a convention as a means of monitoring a commitment. A convention specifies under what circumstances a commitment can be abandoned and how an agent should behave both locally and towards others when one of these conditions arises [16].

The model clearly distinguishes between two concepts: a commitment and a convention. A commitment is a pledge or a promise where a convention is a means of monitoring a commitment, a convention specifies under what circumstances a commitment can be abandoned and how an agent should behave both locally and towards others when one of these conditions arises. An agent commits either to a particular course of action, or, more generally, to a state of affairs. The model is concerned only with commitments that are future-directed towards a state of affairs. Commitment persistence is one of the important features of commitments. By commitment persistence, an agent that has adopted a commitment shall not drop it until it becomes redundant for some reason [16].

As a group of agents engage in a cooperative activity they must create a joint commitment to the overall aim, as well as their individual commitments to the individual tasks assigned to them. The joint commitment shares the persistence property of the individual commitment with the exception that its state is distributed amongst the team members. As a mean to minimize distribution potential drawbacks an appropriate social convention must be defined. The social convention identifies conditions under which the joint commitment can be dropped, and describes how an agent should behave towards its fellow team members. In that context, social conventions act in that context as general guidelines and a common frame of reference in which agents can work. The Adoption of conventions allows every agent to know what is expected both of it, and of every other agent as part of the collective working towards achieving the goal, and knows that every other agent has a similar set of expectations [16].

2.3.3.3 A Four Stages Model for the Cooperative Problem Solving Process:

The CPS process consists of four stages [16]:

Recognition

CPS begins when some agent in a multi-agent community has a goal, and recognizes the potential for cooperative action with respect to that goal due to agent’s belief that it cannot achieve the goal in isolation, or due to its preference for cooperative action as a means to achieve it. This belief it is not enough in itself to initiate the social process. For there to be potential for cooperation with respect to an agent's goal, the agent must also believe there is some group of agents that can actually achieve the goal [16].

Defining the Potential for cooperation

With respect to agent’s goal, there is potential for cooperation iff [16]:

1. There is some group such that the agent believes that the group can jointly achieve goal
2. The agent cannot achieve goal in isolation; or
3. The agent believes that for every action that it could perform that achieves goal, it has a goal of not performing that action.

Team Formation

As rational agent identifies the potential for cooperative action with respect to one of its goals, it will attempt to solicit assistance from a group of agents that it believes can achieve the goal. If the agent is successful, then at the conclusion of this team formation stage, the agent will have brought about in such a group a mental state wherein each member of the group has a nominal commitment to collective action. At this stage, an agent group have not yet fixed an action to perform and do not have any kind of commitment other than the commitment to the principle of joint action. Agents have not yet shared and will not share any kind of commitment other than commitment to joint intention. There is no guarantee that team formation should succeed, an agent should only attempt to [16].

The team formation stage can then be characterized as the following assumption about rational agents: an agent *i*, who believes that there is potential for cooperative action with respect to its goal *p*, will eventually attempt to bring about in some group *g*, which it believes can jointly achieve *p*, a state wherein [16]:

1. it is mutually believed in *g* that *g* can jointly achieve *p*;
2. it is mutually believed in *g* that every agent in *g* is individually committed to *p*, relative to *i* still having a goal of *p*; or, failing that, to at least cause in *g*
3. the mutual belief that *i* has a goal of *p*; and
4. The mutual belief that *i* believe *g* can jointly achieve *p*.

Parts (1) and (2) of this definition represent the minimal commitment that the group has towards *i*'s goal *p* if *i* is successful in its attempt to solicit assistance. This commitment does not yet involve a collective goal; merely the mutual belief that the group can bring about the goal, and that every member of the collective is individually committed to the goal on *i*'s behalf [16].

The team formation stage succeeds iff it is mutually believed in *g* that (i) *g* can jointly achieve *p*; and (ii) every agent in *g* has a commitment to *p*, relative to *i* still having a goal of *p*. If team formation is successful, then for the first time there will be a social commitment by a group of agents on behalf of another agent, a nominal commitment to collective action [16].

Plan formation

At this stage, the newly formed collective attempts to negotiate a joint plan. Upon success of plan formation phase the team will have a full joint commitment to the joint goal, and group *g* will have agreed to the means by which it will pursue this joint goal [16].

Execution

Upon success of the previous stage, agents play the roles they have negotiated in the previous stage. Group g will remain committed to mutually believing they are about to perform the action, and then performing it. If ever a team member comes to believe, for example, that agent *i* no longer has a goal of *p*, then this agent will make the team aware of this, and team action will end [16].

2.4 General Models of Teamwork

Next, we introduce two general models of teamwork. In 2.4.1 we introduce the GRATE model of teamwork, and in 2.4.2, we introduce STEAM. The GRATE model is based on the Joint Responsibility Framework [17], while the STEAM model is based on the theory of Joint Intentions, and it parallels and in some cases borrows from the SharedPlans theory [12].

2.4.1 GRATE

GRATE is a general framework that enables the construction of multi-agent systems for the domain of industrial process control. During system building, framework's inbuilt knowledge related to cooperation and control can be utilized. In constructing multi-agent systems, the role of configuring preexisting knowledge is an integral component, where the developer augment built-in knowledge if necessary with domain specific information [17].

A GRATE agent consists of two clearly identifiable components. The cooperation and control layer is the first component, while the second layer is the domain layer. The domain level system is either preexisting or purpose built and is used to solve problems such as detecting disturbances in the electricity network, locating faults and proposing remedial actions. In order to ensure that its activities are coordinated with those of others within the community, the cooperation and control layer is a meta-controller that operates on the domain level system [17].

Communities created using GRATE have a flat organizational structure. The distribution of control among community members allows an agent to have the role of a team member acting in a community of cooperating agent, and to have the second role of an individual. As a result, the community can pursue more than one goal [17].

In order to build applications using the GRATE framework, the application builder provides appropriate domain-specific rules. Applications could be built very rapidly because much of the general domain behavior is already defined. Selecting a subset of the available knowledge for the problem at hand may be part of the configuration process. The configuration may also involve fine tuning of the control strategies of the problem solving modules. This approach has the following advantages over conventional means of constructing multi-agent systems [17]:

- the reuse of problem solving components that leads to increasing reliability and decreasing risk
- decreasing development time since (some knowledge acquisition and coding has already been carried out
- making an effective use of specialists
- It also follows the lead of other disciplines that engineer complex artifacts, such as planes and cars, in that product development would consist predominantly of assembling components.

2.4.2 STEAM

STEAM is a general model of teamwork that enables a team of agents to act coherently in a way that overcomes the uncertainties of complex, dynamic environments in which team members often encounter differing, incomplete and possibly inconsistent views of the world and mental state of other agents [12].

In order to act coherently, team members must flexibly communicate to avoid mis-coordination, and since such environments can often cause particular team members to unexpectedly fail in fulfilling responsibilities, or to discover unexpected opportunities, teams must be capable of monitoring performance, and flexibly reorganizing and reallocating resources to meet any contingencies. [12]:

STEAM is based on the joint intentions theory. In addition, it also parallels and in some cases borrows from the shared plans theory. The joint intentions is used as the basic building block of teamwork, while team members build up a complex hierarchical structure of joint intentions, individual intentions and beliefs about others' intentions based on the shared plans theory [12].

The joint intentions provide STEAM a principled framework for reasoning about communication providing significant flexibility. Commitments drive communication as team members attempt to achieve mutual belief as they build and breakup joint intentions [12].

STEAM also facilitates monitoring of team performance by making use of explicit representation of team goals and plans, enabling team reorganization in case that the individuals responsible for particular subtasks fail in fulfilling their responsibilities, or if new tasks are discovered without an appropriate assignment of team members to fulfill them. Team's joint intentions drive reorganization as well as recovery from failures in general [12].

The following are STEAM capabilities provided by the Joint Intentions theory [12]:

- Joint intentions are used as building blocks of teamwork, creating a hierarchy of mental attitudes of individual team members
- A joint intention leads to an explicit representation of a team activity and hence, facilitating reasoning about team activity

The following is STEAM capabilities provided by the shared plans theory [12]:

- Analogous to partial shared plans, an agent builds up a snapshot of the team's mental state via joint intentions
- Regarding coherence in teamwork, team members must pursue a common solution path in service of their joint intention for the high-level team goal since pursuing alternative paths might cause paths to cancel each other
- Shared plans mandates mutual belief in a common recipe that might be a partial recipe, and shared plans for individual steps in common recipe, generating a recursive hierarchy to ensure coherence

Chapter 3
Proposed Multi-Agent Collaboration Framework

In this chapter, we propose a framework for multi-agent collaboration. In section 3.1, we start by defining the scope and objective of the proposed framework and based on it, we define the proposed methodology in section 3.2. Section 3.3 defines a state model of teamwork. Section 3.4 defines team's conversational model, and finally in section 3.5, we discuss the effect of the framework on MAS architectures.

3.1 Framework Scope and Objectives

In this section, we define the scope of the proposed framework. We first discuss the characterization of collaboration as the development and maintenance of a shared mental state.

3.1.1 The Development of a Shared Mental State

A team of collaborating agents constructs and maintains shared mental states that emerge through ongoing phases of cooperative problem solving. The shared mental state consists of the following set of shared knowledge structures:

a. a dependency graph of achievement goals.

b. a dependency graph of commitments to achieve these goals,

c. a dependency graph of actions believed to achieve these goals,

d. a dependency graph of commitments to these actions,

e. a dependency graph of intentions of actions agents are committed to achieve, and

f. a dependency graph of mutual beliefs about goal relevance and achievement status, status of commitments, status of intentions, and status of actions.

These knowledge structures provide enough abstraction for an individual and for a collective to reason about teamwork. An agent should be able to individually reason and interact at such level of abstraction in order to actively be a member of teamwork.

The creation of knowledge structures is not enforced to occur in sequence. For example, a constraint that a goal graph should be created before commitment hierarchy is built is an invalid one since at some problems, agents might interleave planning and execution. On the other hand, a valid assumption is that the creation of graphs should take into considerations the dependencies enforced.

3.1.2 The Lack of an Open Collaboration Framework

There are recent trend to integrate agent technology with the World Wide Web and e-Marketplaces [18, 19]. An open system is a system composed of a variable number of interacting components. Interacting parts own the following characteristics [1]:

- they are typically developed independently; i.e. agents are heterogeneous
- parts act concurrently and asynchronously
- there is decentralized control
- parts have limited and potentially inconsistent views of the overall system.

The agent paradigm provides a higher level of abstraction than that provided by the object-oriented paradigm, making it more suitable for the development of open software systems that deal with complex, heterogeneous, and unpredictable environments, with system components of high level of heterogeneity [20]. Using this approach, an open system is modeled as an organization of a set of heterogeneous agents interacting within a multi-agent system (MAS). The use of the agent paradigm to support open systems results in diversity in platforms, architectures, and interaction protocols. This diversity affects the approach used to construct a general framework for collaboration in the sense that the previous assumption of a specified agent architecture, organization or interaction protocols should constrain framework openness. For example, the STEAM framework is built on top of the SOAR environment [12]. In addition, for agent to cooperate in a team, the GRATE framework enforces specific agent architecture and assumes a flat team organization [17]. Finally, the two frameworks, GRATE and STEAM assume a specific pre-specified interaction protocols.

We conclude that a successful collaboration framework supporting an open environment should focus on helping team agents maintain shared mental state, independent of the form of negotiation, coordination and cooperation, agent organization, or distributed problem solving techniques in use.

The scope of the framework can be defined as creating, sharing, and maintaining a shared mental state within a team of agents. Given this scope, we assume the existence of some protocols and mechanisms for negotiation, coordination, cooperation, decision making, and so on. The proposed framework assumes that as agents negotiate, coordinate, and cooperate, they should be exchanging information about their goals, beliefs, commitments, actions and intentions. In addition, we also assume that agents are obeying some rules governing their behavior that allows agents to unambiguously reason about. The proposed framework functionality does not intersect with existing agent development tools and technologies for communication, interaction and knowledge.

3.1.3 Desiderata for an Open Collaboration Framework

The following list represents a desiderata for an open multi-agent collaboration framework:

1. A framework should not enforce a specific type of agent architecture or internal knowledge representation.
2. A framework cannot assume the usage of a specific interaction protocol for cooperation, coordination, negotiation.
3. A framework cannot assume the use of a designated development environment.
4. The framework should focus on being a specification that agents should follow rather than to be in the form of code modules. This form would allow the specification to be continuously enhanced and standardized. In this way, the framework acts as a social law, characterizing agent's social behavior that agents in a MAS would follow as a model.
5. The framework should take into consideration the existence of standards like FIPA, existing development tools, and existing MAS environments, which defines an implementation level specification of protocols, knowledge structures, agent communication languages, and so on.
6. The framework should focus on maintaining a shared mental state within team, which characterize collaborative behavior.
7. The framework should focus on strategies to maintain the team's shared mental state, a characteristic of collaborative behavior. In addition, the framework should propose generic strategies for recovering from teamwork failures.
8. The framework should provide enough verification that enables team agents to reason about both team and individual actions.

9. The framework should provide necessary mechanisms for team agents in order to effectively and clearly contribute to collaborative interactions.
10. The framework specification should be at a level higher than the level of an implementation. This approach allows an abstract view of the framework that framework implementers could then use to map into the various types of open environments.
11. The framework should be build on top of a general formal theory of teamwork of wide acceptance. This approach verifies the generality of the proposed framework.

3.2 The Methodology

In this section, we introduce a proposed methodology to the development of an open collaboration framework.

The proposed methodology is based on the observation that behavior can be analyzed without any knowledge of the implementation details. Given this view, the agent behavior is seen as an external specification for the agent and hence, behavioral models are independent of the architecture, although certain architectures lend themselves more than others to bringing about certain behaviors [2]. The methodology for developing an open multi-agent collaboration framework is based on the development of a high level specification for agent collaborative behavior that can be mapped to lower level specifications of implementation.

The proposed framework is based on two formal models. The first formal model is the cooperative problem solving process [16]. The second formal model is Joint Intentions theory as defined in [14].

In order to support that scope defined above we need to:

(i) define a pattern of collaborative interaction for information exchange that agents should follow,

(ii) define unambiguous rules for reasoning about agent and team behavior, and

(iii) maintain a clear separation between the generic specification defined by the framework and *possible* implementations of that framework.

The proposed framework defines a specification consisting of the following components:

a. A state model defining possible team states and possible conditions for transition between these states.
b. A layered agent conversation model, in which conversations are defined through conversational patterns consisting of some structured textual format rather than Petri-nets. In addition, the model defines speech acts, query interaction protocol, conversations, and a strategy for conversation.

In addition to the specification, we propose an implementation level realization of the framework, consisting of the following components:

a. An agent communication language supporting knowledge exchange through speech acts
b. A behavior representation language (BRL) for describing agent intentional states, supporting modal and temporal operators and providing an extension mechanism for supporting additional semantic attributes, defined through well defined grammar
c. A message interchange format for exchanging messages in XML (eXtended Markup Language) with associated mapping between BRL constructs and XML elements, and supporting the BRL extension mechanism

3.3 Teamwork State Model

The framework defines a state model consisting of a set of possible team states and a set of possible state transitions. The state model provides team agents with enough verification for reasoning about the team state by defining constraints for moving from one team state to another. Figure 1 outlines possible team states and common transitions between these states.

The state model defines the following team states:

Pre-Planning

Cooperative problem solving starts when an agent, the problem initiator, recognizes the problem, the team then enters into the pre-planning state. At this state, there have been recognition of the problem, but the conditions that govern team formation have not been satisfied yet. The team moves to the planning team state in case that there is (i) a mutual belief that the team is able to achieve goal and (ii) a joint nominal commitment to collective action.

Planning

At this state, team members create a plan for action. Agents announce their individual and joint commitment to planned sub-actions. The team leaves this state in either of two cases, the first case occurs if the mutual belief that the formed team can achieve designated goal has been dropped. The second case occurs if agents have a joint commitment to the main action that is assumed to achieve designated goal. In order for agents to jointly commit to a given group action, there should be individual and joint commitments to all of its sub-actions. Team exits the planning state if (i) team formation conditions discussed below have been violated, causing the team to move to the pre-planning state, (ii) team has reached a joint commitment to a complex action in order to achieve team goal causing the team to move to the post-planning state.

Post-Planning

At this state, team agents have committed to a given joint action. The team exits post-planning state if (i) team drops its joint commitment to complex action, causing the team to move to the preplanning state, or (ii) team has reached a joint intention to execute complex action, causing the team to move to the execution state, or (iii) team formation conditions discussed below have been violated.

Execution

At this state, the team members have jointly intended a designated joint action. The team group or a subgroup might jointly intend a joint sub-action for execution. In addition, an individual team agent might individually intend an action as part of team group or subgroup. The team exits execution state if (i) team drops its intention to execute complex action, causing the team to move to the post-planning state, or (ii) team reached a mutual belief that the goal has been achieved, unachievable, or irrelevant, or (iii) team formation conditions discussed below have been violated, or (iv) team planning conditions discussed below have been violated.

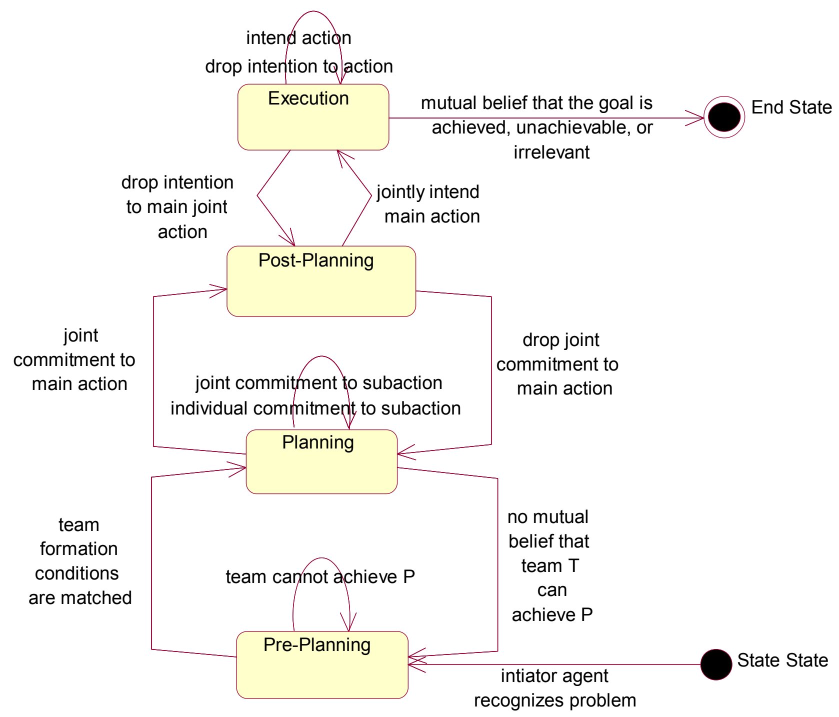

Figure 1: State chart diagram of the cooperative problem solving process

In some situations an agent disjoins team while the during the planning, post-planning or execution team states. For example, if an agent belief that the team goal has been achieved, unachievable, or irrelevant, breaking down mutual belief that the team can achieve designated goal and hence, invalidating all subsequent beliefs, goals, commitments and intentions. In such situations, where a joint mental state is being deformed, agents might start a negotiation to recover this deformation in joint mental state. In the multi-agent trade case study, if team agents reach a mutual agreement that remaining team agents are able to proceed with cooperative problem solving, then they would have retained their joint mental state. If agents fail to recover deformed joint mental state then the team state should rollback to the last known joint mental state.

The framework sets a constraint on team agents to try recovering deformed joint mental state through conducting negotiation before concluding that the joint mental state has been deformed. In the previous example, if negotiation between team agents fails to recover mutual belief, then agents should reason that the team state has changed.

3.4 Teamwork Conversational Model

We define a model of teamwork conversations assuming the approach defined in [6] and described in detail in subsection 1.2.4 – Conversational Patterns.

3.4.1 Speech Acts

The framework defines two speech acts, the *query* speech act and the *inform* speech act.

The Query Speech Act

Using this speech act, an agent sends a message to another agent or to a group of agents querying their mental states in regard to a queried mental element. For example, a team leader agent might send a message to a team member agent requesting it to express its state of commitment to the main goal.

The Inform Speech Act

An agent sends a message to an agent or a group of agents expressing its mental state in regard to a designated mental element. For example, an agent might send a message to all team members to announce that it has dropped its commitment to the main goal.

3.4.2 The Query Interaction Protocol

An interaction protocol is a pattern of sequences of expected communicative acts organized into roles which represents the identity of a single sequence of acts executed by a single entity [6]. The proposed framework defines the query interaction protocol for querying the status of a set of agents. Figure 2 describes the usage of the interaction protocol through an example interaction between an initiator agent and two participants. Two agent roles exist in interaction, the initiator agent and one or more participating agents. The interaction protocol starts when the initiator agent sends a query mental state message to a set of participating agents. Next, every agent receiving the query message then replies back by sending announce mental state message to every agent in the reply-to message field.

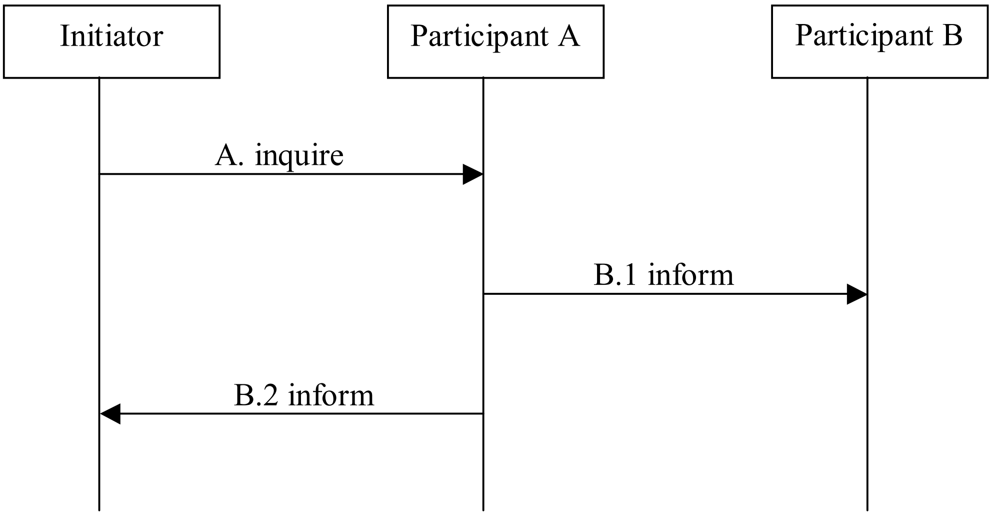

Figure 2: A conversation representing query interaction

3.4.3 Agent Conversational Patterns

We argue that as agents interact, they iteratively build up a conversational context by matching current conversation preconditions and message sequences with pre-conditions and message sequences defined by the framework model conversational pattern. For example, if a team agent A receives an announcement that team member B has committed to a collective action, and given the precondition that there is no joint commitment to this collective action and that the team is in the planning team state, the team agent A would reason that team is contributing to a conversational pattern of reaching a joint commitment to a designated action.

A conversational pattern is valid only at a set of team states. As an example, if the team were in the team planning state, then the form team conversational pattern would not be a valid one. Each of these conversational patterns consists of steps that execute in sequence.

Form Team Conversational Pattern

This conversational pattern defines the preconditions, post-conditions, reasoning, and mental states used to form a team.

Valid Team States

Pre-Planning

Pre-Conditions

- the team formation facilitator agent has a designated goal to achieve through team action.

Step 1: Reasoning Performed by the Team Formation Facilitator Agent

a. belief that a candidate team is able to achieve designated goal.

b. attempt to solicit assistance in order to form team to achieve designated goal.

Step 2: Messages Sent by the Team Formation Facilitator Agent

c. announce to candidate team agents its attempt to solicit assistance in order to achieve designated goal.

Step 3: Reasoning Performed by Other Candidate Agents

d. maintain belief about team formation facilitator announced attempt.

Assumption

- negotiation might be performed at this stage. for example, in the trade domain, negotiations of price, payment terms and delivery method takes place.

Step 4: Reasoning Performed by All Candidate Agents

e. individually reason that candidate team is able to achieve designated goal.

Step 5: Messages Sent by the Team Formation Facilitator Agent

f. announce to candidate team agents its belief that candidate team can achieve designated goal.

g. sends a query mental state to all the candidate team agents, inquiring their belief regarding candidate team's ability to achieve designated goal, requesting reply to be sent to all candidate team agents.

Step 6: Messages Sent by the Other Candidate Agents

h. each agent replies to all other candidate agents announcing its beliefs about team's ability to achieve goal.

Step 7: Reasoning Performed by All Candidate Agents

i. decide whether there is a mutual belief regarding team's ability to achieve designated goal.

Step 8: Reasoning Performed by All Candidate Agents

j. if a mutual belief has been reached, the agent commits to nominal action relative to a designated goal and as part of candidate team.

Step 9: Messages Sent by the Team Formation Facilitator Agent

k. announce its commitment to nominal action in order to achieve designated goal.

l. sends a query mental state to all candidate agents, inquiring their commitment to nominal action in order to achieve designated goal, requesting reply to be sent to all candidate agents.

Step 10: Reasoning Performed by All Candidate Agents

m. in case a joint commitment has been reached, agents individually reason that team state has changed to the planning state.

Post-Conditions

- team has failed to construct team
- team has been successfully formed, team state is planning state

Jointly Commit to a Designated Joint Goal Conversational Pattern

this scenario defines agent conversation used to reach a joint commitment to a designated joint goal and relative to some condition.

Valid Team States

Planning

Pre-Conditions

- a group of agents, which our designated group is a subset of, have reached a joint commitment to parent goal, if any.
- the initiator agent beliefs that this goal is not achieved yet.
- the initiator agent has a given designated goal as an achievement goal.
- there is no joint commitment within this group of agents to achieve that designated goal.

Step 1: Messages Sent by the Initiator Agent

a. announce to other group agent that it is committed to achieving that designated goal relative to some condition.

Step 2: Reasoning Performed by Other Group Agents

b. each agent in the group, other than the initiator agent, will update its mental state to reflect its belief about initiator agent's commitment to designated goal.

Step 3: Messages Sent by the Initiator Agent

c. query other group agents' mental states regarding the status of commitment to that designated goal.

Step 4: Messages Sent by the All Group Agents

d. each agent in the group, other than the initiator, sends a single message to all group agents announcing its mental state in regard to its commitment to achieving the designated goal as part of a group.

Step 5: Reasoning Performed by All Group Agents

e. based on the received feedback, each member of the group would then update its mental state to reflect its beliefs about other agents' individual commitments to designated goal.

f. each agent then, would reason about the joint commitment to achieve designated goal.

Reach Joint Commitment to a Designated Joint Action Conversational Pattern

this scenario defines agent conversation used to reach a joint commitment to a designated joint action.

Valid Team States

Planning

Pre-Conditions

- our designated agent group, or a subset of, has committed to all sub-action of that action, if any exists. sub-actions are either individual or joint actions.
- the initiator agent beliefs that a designated action can achieve a given designated goal which is not achieved yet. moreover, there is a joint commitment to that designated joint goal.
- there is no joint commitment within this group of agents to designated joint action.

Step 1: Messages Sent by the Initiator Agent

a. announce to other group agent that it is committed to this designated joint action relative to some condition.

Step 2: Reasoning Performed by Other Group Agents

b. each agent in the group, other than the initiator agent, will update its mental state to reflect its belief about initiator agent's commitment to designated joint action.

Step 3: Messages Sent by the Initiator Agent

c. query other group agents' mental states regarding the status of commitment to designated joint action.

Step 4: Messages Sent by the All Group Agents

d. each agent in the group, other than the initiator, sends a message to all group agents announcing its mental state in regard to its commitment to group agents announcing its mental state in regard to its commitment to designated joint action.

Step 6: Reasoning Performed by All Group Agents

e. based on the received feedback, each member of the group would then update its mental state to reflect its beliefs about other agents' individual commitments to designated joint action.

f. each agent then, would reason about the joint commitment to designated joint action.

Reach Joint Intention to Execute a Designated Joint Conversational Pattern

This scenario defines agent conversation used to reach a joint intention to execute a designated joint action.

Valid Team States

Post-Planning, Execution

Pre-Conditions

- a group of agents, that our designated group is a subset of, has intended the execution of the parent action, if any.
- there is a joint commitment within this group of agents to execute designated joint action.
- there is no joint intention within this group of agents to execute designated joint action.

Step 1: Messages Sent by the Initiator Agent

a. Announce to other group agent that it has intended the execution of this designated joint action.

Step 2: Reasoning Performed by Other Group Agents

b. Each agent in the group, other than the initiator agent, will update its mental state to reflect its belief about initiator agent's intention of executing designated joint action.

Step 3: Messages Sent by the Initiator Agent

c. Query other group agents' mental states regarding the status of intention to executing designated joint action.

Step 4: Messages Sent by the All Group Agents

d. Each agent in the group, other than the initiator, sends a message to all group agents announcing its state of intention to execute designated joint action.

Step 5: Reasoning Performed by All Group Agents

e. Based on the received feedback, each member of the group would then update its mental state to reflect its beliefs about other agents' individual states of intention to execute designated joint action.

f. Each agent then, would reason about the state of joint intention to execute designated joint action.

3.4.4 The Conversation Policy

We adopt the following definition for the conversation policy [6]:

"...a collection of rules and interaction specifications that guide a particular path or trajectory in a conversation space"

the proposed agent conversation policy consists of the following components:

- Domain and problem specific rules to be defined by the agent developer.
- Teamwork rules explicitly defined by the proposed framework state model through the definition of possible team states and the rules for reasoning about team states.
- Teamwork rules defined by the CPS model of [16]
- Teamwork rules defined by in [14]

3.5 The Effect on Multi-Agent Systems Architectures

Using the proposed framework, two typical MAS patterns exist. The first pattern assumes that a number of agents use the same framework implementation as the uniform mechanism of achieving collaboration between a number of agents. In such a situation, agents might be using the one or more internal form for representing their knowledge. A mapping between each internal format and the language used by the framework implementation is required. In such a case, a facilitator agent may exist for each type of internal representation to provide translation services. In addition, the facilitator should be able to encode and decode messages using the framework implementation's message format, see figure 3.

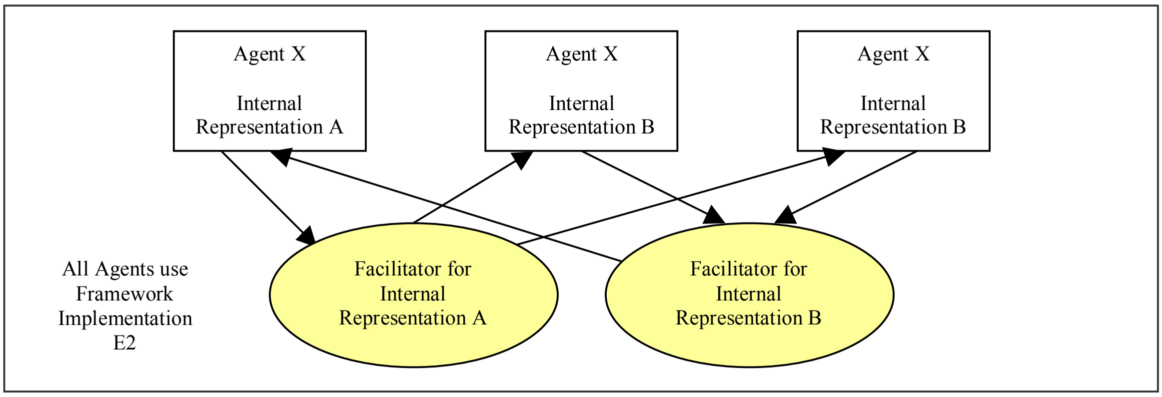

Figure 3: Facilitators translate between internal representations and the framework BRL

The second case occurs if agents of MAS use different framework implementations. In such case, facilitators are required to translate between BRLs, ACLs, and Message formats of different framework implementations, see figure 4.

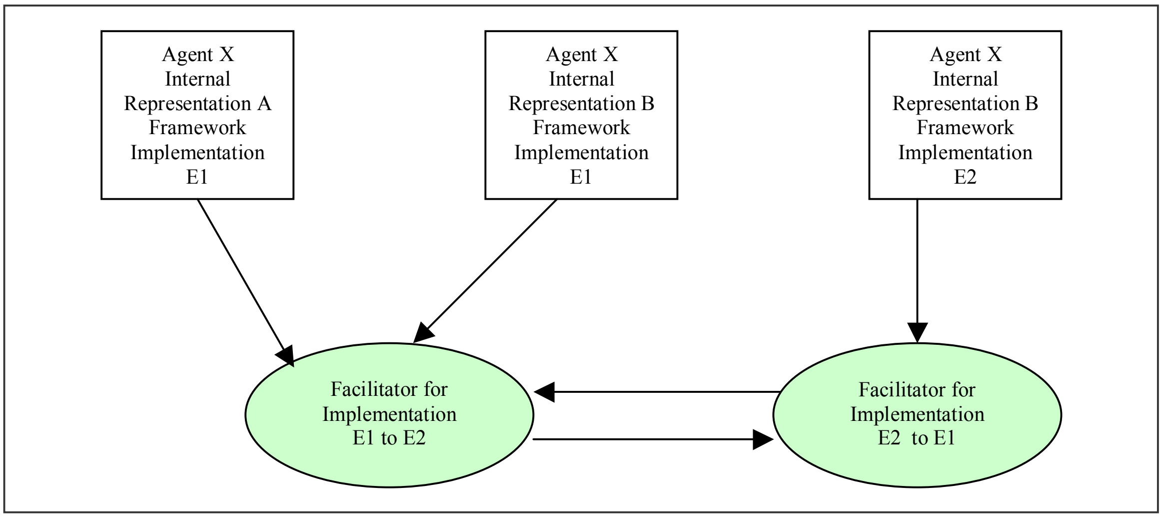

Figure 4: Facilitators translate messages between framework implementations E1 and E2

Chapter 4
Framework Implementation

In this chapter, we propose a framework implementation based on the conceptual model defined in the previous chapter. The proposed framework implementation is one of a set of possible framework implementations, varying in their component definitions. A possible framework implementation would consist of the following components:

- A behavior representation language for modeling agent mental behavior
- An agent communication language supporting the query and inform speech acts
- A message interchange format
- A set of facilitator agents and/or software components, which serve as enablers for that implementation. For example, as we have seen in figure 4 in chapter 3, a translator facilitator agent might be used to translate between implementation E1 and implementation E2.

In section 4.1 of this chapter, we discuss a developed behavior representation language for modeling agent mental behavior. In section 4.2, we describe a developed gent communication language. In section 4.3, we describe a developed message interchange format and in section 4.4, we describe the development of an XML Message Encoding/Decoding Facility.

4.1 Behavior Representation Language (BRL)

The behavior representation language (BRL) is used to exchange information about agent behavior. Following, we describe a proposed approach to define the language, the knowledge structures, language semantics, and XML mapping. BRL assumes an agent with a second-order intentional system. A second-order intentional system is defined to have intentional states about intentional states of itself and about other agents' intentional states [21].

4.1.1 The Knowledge Structures

The object model in figure 5, defines possible relationships between knowledge elements.

All knowledge elements are classified as subclasses of the class named *Mental Element*. The language defines several types of mental elements namely, *Belief*, *Goal*, *Commitment*, *Action* and *Intention*. A mental element can be qualified by a modal or temporal operator. The definition of elements and their associated relationships reflects those defined by the two models CPS [16] and the joint intentions theory [14]. For example, an agent might be committed to achieve a designated goal or to execute a designated action. An intention might be associated with a given action. An agent might adopt a belief about the existence of a designated mental element. An agent might have a goal to achieve relative to some condition defined through the existence of a designated mental element. Table 1 below defines the possible attributes for each of the knowledge elements. The definition of additional semantic attributes to express additional semantics follows the rules we have defined in 4.1.6.

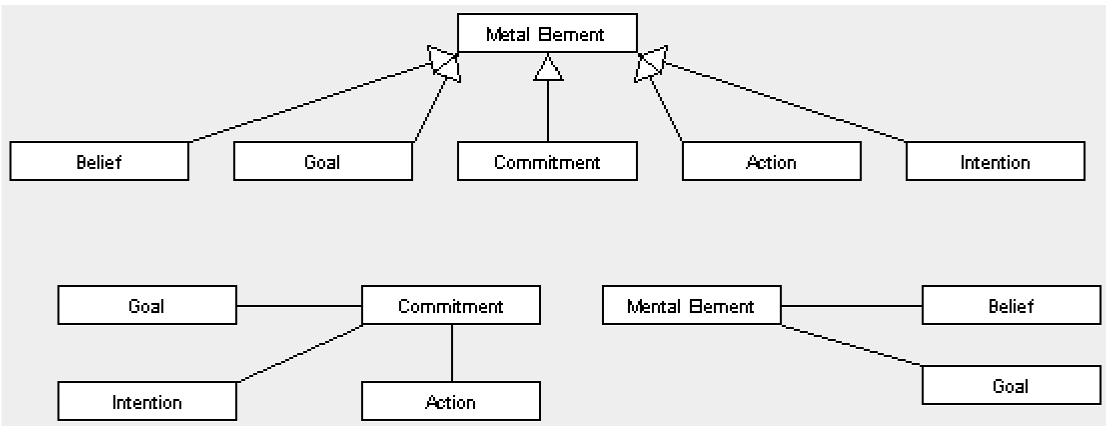

Figure 5: A class diagram showing possible relationships between knowledge elements

Frame	Field Name	Description
Belief	*belief id*	unique identification
	agent id	unique identification for agent or team that owns this belief
Belief	*Message clause id*	unique identification for the message clause, see next section
Goal	*goal id*	unique identification
	agent id	unique identification for agent or team that owns this belief
Commitment	*commitment id*	unique identification
	agent id	unique identification for agent or agent group that owns this commitment
	goal id or action id	unique identification for goal (action) that an agent is committed to achieve (execute)
Action	*action id*	unique identification
	agent id	unique identification for agent or agent group that owns this action
	goal id	unique identification for goal that an agent is committed to achieve through the execution of action
Intention	*intention id*	unique identification
	agent id	unique identification for agent or agent group that owns this intention
	commitment id	unique identification for commitment that an agent is intending

Table 1: Knowledge attributes for each knowledge element

4.1.2 Modal and Temporal Operators

In order for agents to express knowledge that is associated with time and possibility attributes, BRL provides supports for such constructs via the support of a set of modal and temporal operators. For example, an agent might express the addition of a belief about the achievability of a designated joint goal, hence allowing other agents to reason about that information and act accordingly. The language supports the following set of modal and temporal operators listed in tables 2 and 3.

Modal clause	Description
achievable	Expresses the status of achievability of a given goal
irrelevant	Expresses the status of relevance of a given goal
Exist	Expresses the status of existence of a designated knowledge element.

Table 2: BRL language modal operators

Temporal clause	Description
Added	Expresses the status of addition of a new designated knowledge element.
Dropped	Expresses the status of dropping a designated belief
achieved	Expresses the status of achievement of a given goal
Attempted	Expresses the past status of attempt in the past of a designated action as a result of an agent intention
attempting	Expresses the current status of attempt in execution of a designated action as a result of an agent intention

Table 3: BRL language temporal operators

4.1.3 Language Clauses

The framework implementation supports two types of language clauses, BDI clauses and message clauses. BDI clauses are used to represent knowledge about mental elements. BDI clauses does not express neither the possibility nor the point of time at which a certain condition is true. Message clauses support modal and temporal relations. Message clauses are used to express and query the possibility and the truth condition at a given point of time. For example, using the exist clause, an agent might express that a given mental element is currently being believed.

An agent associates its belief with a language operator expression through its id. This allows an agent for example to belief that another agent has dropped a designated goal. Each of operators above has two attribute named *mental element id* and *truth* value. The *mental element id* defines associated knowledge element id. The *truth value* attribute defines the truth value of operator expression and contains either a bounded value of ***true*** or ***false***, or a variable id, containing some dummy identifier. An operator expression with bounded value, either true or false, states the truth value of a knowledge element, while the use of an unbounded value states a query about the truth of a knowledge element that needs to be resolved.

4.1.4 A Grammar for BRL

We define a grammar for BRL language, see figure 6, assuming a non-structured textual representation of BRL clauses that allows nesting of agent clauses. The grammar is intended as a mean to describe syntactical and semantic aspects of language clauses. XML notation, see 4.3, defines a language clause as a separate XML node with the list of identifiers represented as an attribute list. Identifiers shall be generated by some system that is outside the definition of the language. Figure 6 shows the grammar for BRL, non-terminals are placed between < and > signs.

<language clause>	::=	<message clause> | <BDI clause>
<message clause>	::=	<added clause> | <exist clause> | <dropped clause> | <irrelevant clause> | <achievable clause> | <achieved clause> | <unknown clause> | <attempted clause> | <attempting clause>
<exist clause>	::=	exist (belief clause id | goal clause id | commitment clause id | intention clause id | attempt clause id, truth value)
<dropped clause>	::=	dropped (belief clause id | goal clause id | commitment clause id | intention clause id | attempt clause id, truth value)
<irrelevant clause>	::=	irrelevant (goal clause id, relevance goal clause id, truth value)
<achievable clause>	::=	achievable (agent id, goal clause id, truth value)
<achieved clause>	::=	achieved (action clause id, truth value)
<attempted clause>	::=	attempted (action id, intention id, truth value) The attempted clause defines the truth of an attempt that happened in the past
<unknown clause>	::=	unknown (message clause id, truth value) The unknown clause is used to express that the enclosed message clause contains knowledge that is not included in the agent's knowledge base. The clause is used in reply to query speech acts.

Figure 6: A Grammar for the Behavior Representation Language (BRL)

<attempting clause>	::=	attempting (action id, intention id, truth

		value) The attempted clause defines the truth of an attempt that is still happening
<BDI clause>	::=	<belief clause> | <goal clause> | <action clause> | <commitment clause> | <intention clause>
<belief clause>	::=	belief(agent id, belief clause id, message clause id) This clause defines a belief of an agent defined by id. the belief is defined in terms of a message clause, enabling the use of modal and temporal operators to represent knowledge. It is to be noted that referenced message clause might be associated directly or indirectly with a belief clause which itself is associated with a message clause
<goal clause>	::=	goal(agent id, goal clause id) This clause defines a goal of an agent defined by id
<action clause>	::=	action(agent id, action clause id, goal id) This clause defines an action that is associated with an agent
<commitment clause>	::=	commitment(agent id, commitment clause id, goal clause id) This clause defines a commitment of an agent to achieve a designated goal commitment(agent id, commitment id, action id) This clause defines a commitment of an agent to execute a designated action
<intention clause>	::=	intention(agent id, intention clause id, commitment clause id) This clause defines the intention to of an agent based on a designated commitment to an action
truth value	::=	true | false | placeholder
agent id	::=	Identifier An identifier for an individual agent or a group of agents

Figure 6, continued: A Grammar for the Behavior Representation Language

belief id	::=	Identifier

```
goal id             ::=  Identifier
team id             ::=  Identifier
nominal action id   ::=  Identifier
action id           ::=  Identifier
team id             ::=  Identifier
commitment id       ::=  identifier
Intention id        ::=  identifier
Identifier          ::=  a..z|A..Z[a..z|A..Z|0..9|_]*
Placeholder         ::=  A..Z[A..Z]*
```

Figure 6, continued: A Grammar for the Behavior Representation Language (BRL)

4.1.5 BRL as an Ontology for Collaborative Mental Behavior

Given the definition of ontology by [1], BRL acts as ontology for the framework implementation in the sense that it is a specification of the objects, concepts, classes, functions, and relationship defining collaborative mental behavior for an agent. The ontology can be shared and reused by many agents speaking common vocabulary of BRL within the domain of collaborative mental behavior.

4.1.6 A Proposed Approach to Define Semantic Attributes

We define the behavior representation language (BRL) in order to represent agent knowledge-level behavior. For example, committing to an action, or dropping a belief is characterized as a knowledge-level behavior. BRL language defines a set of clause types.

Our approach to represent language constructs is through the encoding of language clauses into a carrier structure. In the proposed framework implementation, the carrier structure is XML. Every language clause has its own mapping into XML node. Using this approach, we represent language grammar in the form of knowledge frames. Each language frame defines grammar elements of a language clause.

Many systems support additional set of semantic attributes for commitments, goals, actions, and so on. For example, one system may have a new semantic attribute for *freshness* of a commitment. In such a representation, a commitment would expire after a given time interval. Another system may support the definition of rules to define the *convention* associated with a given commitment.

As a result, the different varieties of attributes cannot be enumerated at the time of framework implementation. For this reason, the implementation should provide the flexibility that would allow MAS implementers to define their own semantics. This flexibility is provided through the use of XML as the carrier format for BRL, and optionally through the use of the implemented XML Message Encoding/Decoding Facility.

4.2 Agent Communication Language

The framework implementation defines an agent communication language for supporting two speech acts namely, inquire and inform speech acts. The language support such speech acts through the definition of two mechanisms to support inform and query acts as defined below.

4.2.1 The Inform Mechanism

An agent that would like to inform a set of agents about the status of its mental behavior would send a message clause with the truth value attribute set to either true or false values. An agent receiving such a message is expected to update its mental state using this piece of knowledge.

4.2.2 The Inquire Mechanism

An agent that would like to inquire the status of mental behavior of an agent would send a message clause with the truth value attribute set to a value other than true or false values. An agent receiving such a message is expected to reply back expressing the truth or falsity of such an expression through sending back the same message clause with the exception that the truth value is set to either true or false.

4.3 Message Interchange Format

Next, we describe a message interchange format that uses XML as a carrier. We start first by defining the rules for mapping language constructs of ACL and BRL defined previously to equivalent XML nodes and attributes. Next, we define an XML document structure.

4.3.1 XML Mapping for ACL and BRL

Language constructs is being mapped to XML corresponding elements that acts as the carrier for language constructs. Each language clause maps to an XML node and each of the clause's semantic elements, pre-defined or user defined, maps to an XML attribute. All XML nodes, representing language clauses, are the sibling of the same node within an XML document hierarchy.

XML node names are the same as of clause and operator names. XML attribute names are the same as those above with white spaces converted into an underscore. Truth values are literals *true* and *false*. The specification of identification values are application defined.

4.3.2 XML Message Structure

An XML message consists of a main node, the team_message node, which contains a meta_info node, a delivery_info node, and a message_content node. The meta_info node defines the framework implementation used. The delivery_info node contains a sender_id attribute, a message recipient_list, and a reply_to_list. The recipient_list and the reply_to_list contain a node for each recipient agent. The message_content list consists of a primary message_clause node and a number of message_clause and bdi_clause nodes. The primary message clause is associated with one primary bdi_clause. The primary bdi_clause is defined through the use of a number of bdi_clause and message_clause nodes. Other message clauses are associated with BDI clauses representing beliefs. Figure 7 shows a sample XML document representing a team message.

```
<team_message team_id="trade_team" message_id="attempt_to_solicit_assistance_1">
    <meta_info content_type="framework-implementation-horn-clause"/>
    <delivery_info sender_id="buyer">
        <recipient_list>
            <recipient agent_id="merchant"/>
            <recipient agent_id="delivery"/>
        </recipient_list>
        <reply_to_list>
            <recipient agent_id="merchant"/>
            <recipient agent_id="delivery"/>
```

Figure 7: A sample team message in XML format

```
        <recipient agent_id="buyer"/>

     </reply_to_list>

  </delivery_info>

  <message_content>

     <message_clause
            truth_value="true"
            bdi_clause_id="SolicitAssistance"
            type="added"
            id="SolicitAssistance"/>

     <bdi_clause
            entity_id="buyer"
            type="goal"
            id="SolicitAssistance_goal"/>

     <bdi_clause
            action_id="SolicitAssistance_action"
            entity_id="buyer"
            commitment_id="SolicitAssistance_commitment"
            type="intention"
            id="SolicitAssistance_intention"/>

     <bdi_clause
            entity_id="buyer"
            type="action"
            id="SolicitAssistance_action"/>

     <bdi_clause
            entity_id="buyer"
            type="attempt"
            intention_id="SolicitAssistance_intention"


            id="SolicitAssistance_attempt"/>
```

```
     <bdi_clause
            entity_id="buyer"
            goal_id="SolicitAssistance_goal"
            type="commitment"
```

Figure 7, continued: A sample team message in XML format

```
              id="SolicitAssistance_commitment"/>
    </message_content>
</team_message>
```

Figure 7, continued: A sample team message in XML format

4.4 XML Message Encoding/Decoding Facility

The framework implementation supports communication through the exchange of XML documents. Encoding and decoding of XML messages is enabled through the use of XML Message Encoding/Decoding Facility.

4.4.1 The Object Model

The object model for the XML Message Encoding/Decoding Facility can be summarized in figure 8. There are two types of clauses, the MessageClause, created using a MessageClauseFactory, representing a agents mental state with a modal or temporal attribute. An agent message contains only one main MessageClause, and a set of secondary message clauses. A secondary MessageClause is being referenced by BDIClauses included in the XML message. A BDIClause is created using a BDIClauseFactory. The BDIClause is not associated with any modal or temporal attribute. There are five types of a BDIClause, the IntentionClause, the BeliefClause, the GoalClause, the CommitmentClause and the ActionClause. Association between Clause objects follows rules for knowledge structures defined in 4.1.1 and 4.1.2.

A TeamMessage is created by a TeamMessageFactory. A TeamMessage object is an aggregation of a MessageMetaInfo, a DeliveryInfo and MessageContent objects. The MessageMetaInfo instance contains information about current framework implementation. The DeliveryInfo instance consists of two AgentList instances, representing the recipients and reply-to lists. The MessageContent instance acts as a container for MessageClauses and BDIClauses.

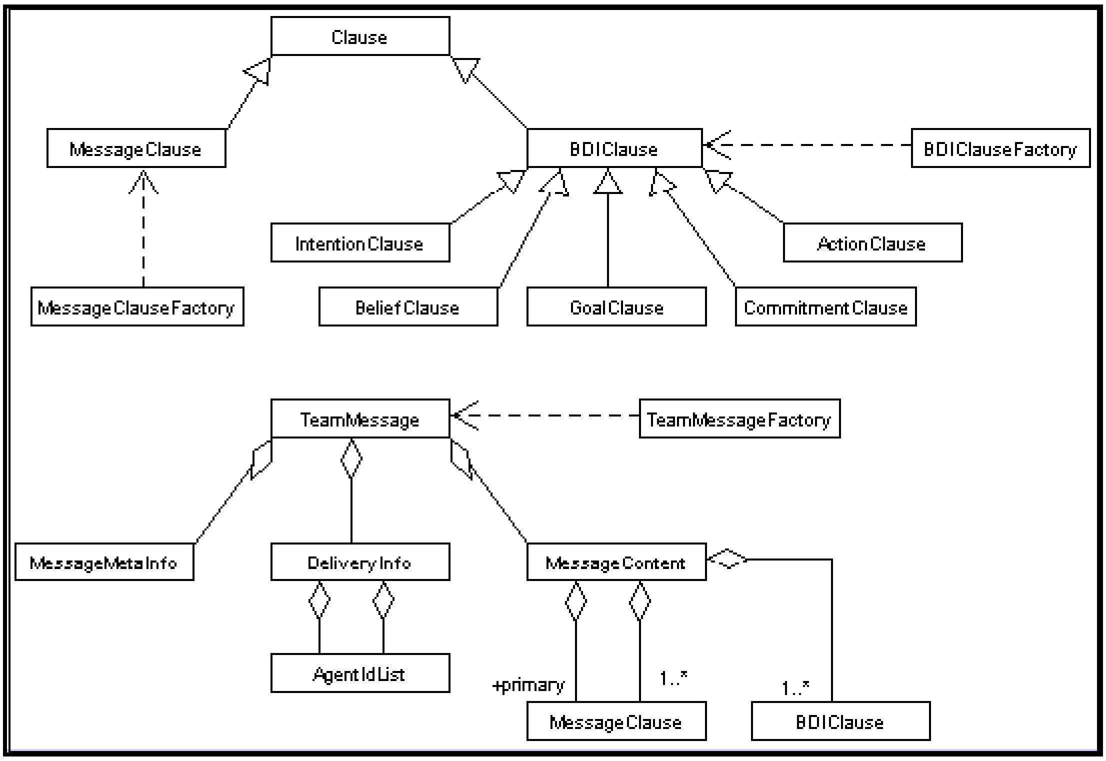

Figure 8: A Static model for XML message encoding/decoding facility

4.4.2 Using the XML Message Encoding/Decoding Facility

Next, we introduce two examples for encoding and decoding messages in XML.

Invocation of the Encoding Function

The following code in figure 9 shows a sample constructs a team message:

```
String      sender = "agentA";
String[]    recipient = {"agentB"};          // a single recipient
                                            // agentB recipient
String[]    replyTo = {"agentA", "agentB"}; // reply to agentA and
                                            // agentB
MessageClause messageClause;
BDIClause      bdiClause;
TeamMessage   tm;
String         xmlMessage;

// 1. create a message content object
MessageContent messageContent = new MessageContent();
// 2. use a factory to create the main BDI clause
```

Figure 9: Invocation of the encoding function

```
bdiClause = BDIClauseFactory.actionIntention(
                                    messageContent,
                                    "buyer",
                                    "intend_solicit_assistance");
// 3. use a factory to create a message clause
messageClause = MessageClauseFactory.addBDIElement(
                                       messageContent,
                                       "solicit_assistance",
                                       "solicit_assistance",
                                       MessageClause.TOKEN_TRUE);
// 4. use factory to construct team message object
tm = TeamMessageFactory.constructTeamMessage("team1",
                                             "message2",
                                             "message1",
                                             sender,
                                             recipient,
                                             replyTo,
                                             messageContent);
// 5. convert team message into an XML document
xmlMessage = TeamMessageFactory.getXMLMessage(tm);
// 6. print the message to standard output
System.out.println("XML MESSAGE" + xmlMessage);
```

Figure 9, continued: Invocation of the encoding function

Invocation of the Decoding Function

The following sample code in figure 10 shows a decoding of an XML team message:

```
String            sender;
DeliveryInfo      dinfo;
MessageClause     messageClause;
MessageContent    mc;
BDIClause         bdiClause;
TeamMessage       tm;
```

Figure 10: Invocation of the decoding function

```
// 1. use XML message string to construct a message object
tm = TeamMessageFactory.getTeamMessage(xmlMessageString);
// 2. obtain a reference to the delivery information object
dinfo = tm.getDeliveryInfo()
// 3. obtain a reference to the sender identification
sender = dinfo.getSenderId();
// 4. obtain a reference to the message content
mc = tm.getMessageContent();
```

Figure 10, continued: Invocation of the decoding function

Chapter 5
Case Study: The Use of MAS in e-Trade

In this chapter, we study the potential for modeling e-Trade entities as a group of collaborating agents with the objective of executing a trade transaction. In this study, we build on top of the proposed framework, and using the proposed framework implementation as the vehicle for the case study implementation. In section 5.1, we introduce the potential of using MAS in e-commerce applications. In section 5.2, we introduce a model of an e-Trade purchase process as the collaboration within a team of trade agents. In section 5.3, we introduce a model for an e-Trade purchase team organization. In section 5.4, we define a knowledge level model for reasoning about collaboration in a trade team performing a purchase. In section 5.5, we describe collaborative interactions occurring within a trade team, with an example of executing a form team interaction scenario. In section 5.6, we introduce the collaborative analysis facility, aiding the specification and analysis of e-Trade team collaborative behavior. Finally in section 5.7, we introduce the case study findings.

5.1 MAS and e-Trade

The term e-commerce stands for a model of modern commerce in which software agents represent the roles of a buyer, a seller, a mediator, a facilitator, and an information provider, resulting into automation of parts or all the business tasks involved in e-commerce. The adoption of e-commerce is believed to result into advantages for the enterprises, in the form of cost reduction, and for the customers, by providing them with better bargaining tools [22].

Software agents can support the following set of commerce activities [23]:

- identifying requirements
- brokering products
- brokering vendors
- negotiating deals
- making purchase and payment transactions

E-commerce applications operates in dynamic and distributed environments, characterized as dealing with a large number of heterogeneous information sources with evolving contents and dynamic availability. E-commerce applications typically rely on distributed and autonomous activities for information search, fusion, extraction and processing, without centralized control. Business partnerships, as found between suppliers, resellers, brokers, and customers, need to be created dynamically and maintained only for the required duration such as a single transaction. Business processes, or workflows, may be conceived as a kind of cooperation between a team of agents in a multi-agent system, where software agents are used to perform tasks of business processes, and workflows orchestrates or control the interactions between agents [23].

5.2 Modeling Purchase as Collaboration among Trade Agents

5.2.1 Understanding the Purchase Process

In a given marketplace, the purchase process is divided into three phases: negotiation, exchange, and then settlement. The first phase starts when the buyer recognizes the need to purchase some merchandise. The buyer then would request an offer from merchandise providers, the merchants, who are known to supply this type of merchandise. Merchants would then reply to the buyer with their offers, varying in their price, quality, payment terms, and delivery conditions. The buyer starts the negotiation process by commenting on these offers, which merchants would then modify and send again to the buyer. The negotiation phase ends as the buyer decides either to select a given offer for purchase, or to drop the purchase decision. After the buyer and the selected merchant would then agree on all the offer details, they move to the exchange phase. For the remainder of this chapter, performing a trade corresponds to performing a trade transaction.

The exchange phase starts when both the buyer and the merchant exchange values, for example: money for merchandise. The buyer agent performs the payment according as agreed upon. The merchant then would deliver the merchandise to the buyer's site.

The settlement phase starts when the buyer and the merchant update their records and declare that the purchase is complete.

5.2.2 Trading Agents as Representatives of Trade Entities

The potential of using intelligent agents lies in their ability to act in an intelligent manner, featuring aspects such as autonomy, learning, and adaptation to environmental changes. For example, the purchase process can be modeled as a team of collaborating trade agents, where each role is associated with one or more agents.

The use of intelligent software agents in e-commerce applications can be divided into two categories. In the first category, intelligent agents are be used to automate some parts of the trading process. For example, in searching the internet for the best offers in terms of price and delivery conditions. The second type of application involves the full automation of e-commerce processes. For example, the automatic reorder of inventory items as a result of firing of an inventory reorder event.

5.2.3 Mapping of Purchase Process Phases to CPS Model Phases

The purchase process can be conceived as a process of cooperative problem solving. Table 4 describes the relationship between CPS model phases and purchase process phases.

5.2.4 Teamwork Aspects in the Purchase Process

Actions of a purchase transaction are required to execute in a given sequence agreed upon by all team agents. Actions of a purchase transaction are either an individual action executed by an individual or joint action executed by a group. The purchase transaction itself is a joint action. Coordination is required between trading entities in order for trade transaction to succeed.

Trading entities should behave as a team in the sense that they succeed or fail together. The transaction is a joint effort between all trade team members, requiring them to reason about collaboration and to act in a manner that benefits the team. For example: if the merchant agent decided not to continue with the purchase, it should announce this to the rest of the trade team members.

Purchase Process Phase	**CPS Model Phase**	**Description**
Decision to buy merchandise	Recognition	The buyer recognize the potential for performing purchase
Negotiation phase	Pre-team	Buyer negotiates with a group of merchants to select the best offer, and then, construct a trade team in order to execute the transaction
	Planning	Buyer and merchant agree on all purchase attributes
Exchange phase	execution	Trade team members execute purchase transaction steps in a timely coordinated manner
Settlement phase		

Table 4: Mapping of Purchase Process Phases to CPS Model Phases

5.3 The Organization of the Trade Team

In this case study, we assume that the organization of a trade team consists of a buyer role, a merchant role, and a delivery role. The buyer and merchant roles are being associated with two different agents. Each role is associated with a number of responsibilities, and performs a number of specific trade actions that we shall discuss later in this chapter. At the remainder of this section, we discuss the responsibilities of each agent role.

5.3.1 Buyer Responsibilities

The buyer role is responsible for representing a buyer in a trade. A buyer role is able to perform payment using a set of payment methods, receive merchandise delivered through the delivery agent, and settle its part of the trade transaction.

5.3.2 Merchant Responsibilities

The merchant role is responsible for representing a merchant in a trade. A merchant role is able to receive payment from the buyer, and to settle its part of the trade transaction.

5.3.3 Delivery Responsibilities

The delivery role is responsible for representing the delivery organization in a trade. A delivery role is able to delivery merchandise from the merchant site to the buyer site.

5.4 A knowledge Level Model for Reasoning about Collaboration in e-Trade

In this section, we define a knowledge model that we use later in our case study. The knowledge model helps trade agents' reason about collaboration within a team of trade agents performing a purchase. The architecture of trade agents used in this case study is based on BDI model. The knowledge model consists of a group of mental elements, beliefs, goals, commitments, and intentions. The model defines the owner and type of each mental element and the addition and drop triggers. The addition trigger defines the conditions that result in the adoption of that element. The drop trigger defines the conditions that result in dropping off the mental element.

Trade team members achieve collaboration by expressing their mental state and querying mental states of others agents using BRL defined in the framework implementation. BRL is being used to define the knowledge model mental elements and their interrelationships.

In 5.4.1, we provide a detailed description of trade agents' goals. In 5.4.2, we provide a detailed description of trade agents' actions. In 5.4.3, we provide a detailed description of trade agents' commitments. In 5.4.4, we provide a detailed description of trade agents' intentions. In 5.4.5, we provide a detailed description of trade agents' beliefs.

In tables 4 through 8, capital letter B denotes the buyer agent, capital letter M denotes the merchant agent, and capital letter D denotes the delivery agent.

5.4.1 Agent Goals

The trade agent achievement goals are defined via table 5 and figure 11. Table 5 defines the goal identifier, owner agent, type, and addition and drop triggers. Figure 11 defines achievement goal hierarchy showing goal dependency.

Goal	Agent	Type	Addition Triggers	Drop Triggers
solicit assistance	B	individual	buyer recognizes the need for performing trade	team has been formed buyer beliefs that perform trade goal is unachievable or irrelevant
perform trade	B, M, D	joint persistent goal	buyer, merchant and delivery individually convinced that the goal is not achieved and is achievable each agent individually validates the state of joint persistence	one or more agents has dropped its goal
deliver merchandise	D	individual	planning	goal achieved, unachievable or irrelevant

Table 5: trade team achievement goals

Goal	Agent	Type	Addition Triggers	Drop Triggers
receive payment	M	individual	planning	goal unachievable or irrelevant
receive merchandise	B	individual	planning	goal achieved, unachievable or irrelevant
make the state of achievability and relevance of perform trade goal known to other team members	B, M, D	individual	as associated persistent goal is added	as associated persistent goal is dropped
settle buyer part of trade transaction	B	individual	as associated persistent goal is added	goal achieved, unachievable or irrelevant
perform payment	B	individual	planning	goal achieved, unachievable or irrelevant
make the state of achievability and relevance of perform payment goal known to other team members	B	individual	as associated persistent goal is added	as associated persistent goal is dropped
settle merchant part of trade transaction	M	individual	planning	goal achieved, unachievable or irrelevant

Table 5 continued: trade team achievement goals

Goal	Agent	Type	Addition Triggers	Drop Triggers
make the state of achievability and relevance of receive payment goal known to other team members	M	individual	as associated persistent goal is added	as associated persistent goal is dropped
make the state of achievability and relevance of deliver merchandise goal known to other team members	D	individual	as associated persistent goal is added	as associated persistent goal is dropped
make the state of achievability and relevance of receive merchandise goal known to other team members	B	individual	as associated persistent goal is added	as associated persistent goal is dropped
make the state of achievability and relevance of merchant commitment of trade transaction goal known to other team members	M	individual	as associated persistent goal is added	
make the state of achievability and relevance of buyer commitment of trade transaction goal known to other team members	B	individual	as associated persistent goal is added	

Table 5 continued: trade team achievement goals

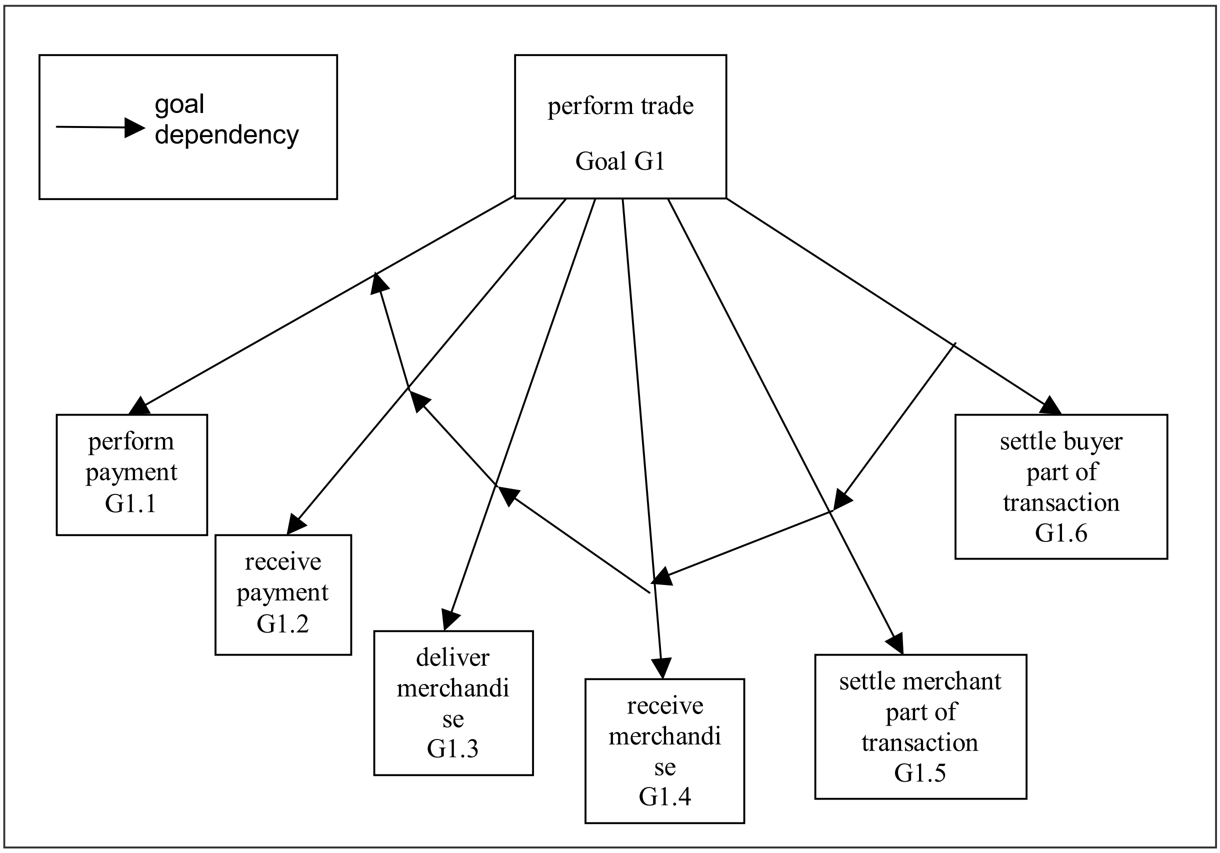

Figure 11: Goal hierarchy showing goal dependency

5.4.2 Agent Actions

The trade agent actions are defined via table 6, which defines the action identifier, owner agent, parent action, action type, and dependency of action execution.

Action	Agent	Parent Action	Action Type	Dependency
perform trade	B, M, D	none	complex	perform payment, receive payment, deliver merchandise, receive merchandise, merchant commit merchant part of transaction, buyer commit merchant part of transaction
perform payment	B	perform trade	individual	none
receive payment	M	perform trade	individual	perform payment
deliver merchandise	D	perform trade	individual	receive payment
receive merchandise	B	perform trade	individual	deliver merchandise
settle buyer part of trade transaction	B	perform trade	individual	receive merchandise
settle merchant part of trade transaction	M	perform trade	individual	settle buyer part of trade transaction

Table 6: trade team agent actions

5.4.3 Agent Commitments

The trade agent commitments are defined via table 7, which defines the commitment identifier, owner, type, and addition and drop triggers.

Commitment	Agent	Type	Addition Triggers	Drop Triggers
commit to nominal action	B, M, D	joint	Agent reasoning	Agent reasoning
commit to perform trade goal	B, M, D	joint	planning	goal is achieved, unachievable or irrelevant
commit to perform payment goal	B	individual	planning	goal is achieved, unachievable or irrelevant
commit to receive payment goal	M	individual	planning	goal is achieved, unachievable or irrelevant
commit to perform trade action	B, M, D	joint	planning	drop commitment to corresponding goal commitment, or action is not believed to achieve goal, or drop commitment to sub-action
commit to settle buyer part of trade transaction goal	B	individual	planning	goal is achieved, unachievable or irrelevant

Table 7: trade team commitments

Commitment	Agent	Type	Addition Triggers	Drop Triggers
commit to deliver merchandise goal	D	individual	planning	goal is achieved, unachievable or irrelevant
commit to settle merchant part of trade goal transaction	M	individual	planning	goal is achieved, unachievable or irrelevant
commit to perform payment action	B	individual	planning	drop commitment to corresponding goal commitment, or action is not believed to achieve goal
commit to receive payment action	M	individual	Planning	drop commitment to corresponding goal commitment, or action is not believed to achieve goal
commit to deliver merchandise action	D	individual	planning	drop commitment to corresponding goal commitment, or action is not believed to achieve goal
commit to receive merchandise action	B	individual	planning	drop commitment to corresponding goal commitment, or action is not believed to achieve goal

Table 7 continued: trade team commitments

Commitment	Agent	Type	Addition Triggers	Drop Triggers
commit to receive merchandise goal	B	individual	planning	goal is achieved, unachievable or irrelevant
commit to settle buyer part of trade transaction action	M	individual	planning	drop commitment to corresponding goal commitment, or action is not believed to achieve goal
commit to settle merchant part of trade transaction action	M	individual	planning	drop commitment to corresponding goal commitment, or action is not believed to achieve goal

Table 7 continued: trade team commitments

5.4.4 Agent Intentions

The trade agent intentions are defined via table 8, which defines the intention identifier, owner, type, and pre and post conditions.

Intention	Agent	Type	Precondition	Drop Triggers
perform trade action	B, M, D	joint	trade team agents jointly intending action	drop intention to a sub-action
perform payment action	D	individual	joint intention to parent action is reached	drop commitment to corresponding action
receive payment action	M	individual	joint intention to parent action is reached	drop commitment to corresponding action
deliver merchandise action	D	individual	joint intention to parent action is reached	drop commitment to corresponding action
receive merchandise action	D	individual	joint intention to parent action is reached	drop commitment to corresponding action
settle merchant part of trade transaction action	M	individual	joint intention to parent action is reached	drop commitment to corresponding action
settle buyer part of transaction	D	individual	joint intention to parent action is reached	drop commitment to corresponding action

Table 8: trade team intentions

5.4.5 Agent Beliefs

The trade agent beliefs are defined via table 9, which defines the belief identifier, owner, and addition and drop triggers.

Belief	Agent	Addition Triggers	Drop Triggers
trade is not performed yet	B, M, D	agent reasoning	the goal of performing trade is known to be achieved
the agent should perform trade	B	agent reasoning	goal of performing trade has been achieved, unachievable, or is irrelevant
trade team, consisting of the buyer b, merchant m, and delivery d agents, is able to perform trade	B, M, D	agent reasoning (buyer) agent negotiation (merchant, delivery)	agent reasoning and agent negotiation
buyer b should solicit assistance from m and d	B	addition of previous belief	drop of previous belief
agent do (not) have a perform trade as a persistent goal	B, M, D	addition of the beliefs that the trade team is able to perform trade and that the agent is able to perform trade	drop of the belief that the trade team is able to perform trade
agent is (not) committed to persistent goal of perform trade	B, M, D	planning	drop of associated persistent goal

Table 9: trade team beliefs concerning teamwork

Belief	Agent	Addition Triggers	Drop Triggers
trade team is in the pre-planning phase	B, M, D	agent reasoning	agent reasoning
agent is (not) committed to nominal action to achieve persistent goal of performing trade	B, M, D	reasoning and negotiation	reasoning and negotiation
a joint commitment to nominal action has (not) been reached	B, M, D	agent reasoning about communication	agent reasoning about communication
there is a mutual belief that there is a joint commitment to nominal action has (not) been reached	B, M, D	agent reasoning about communication	agent reasoning about communication
an agent has committed (dropped commitment) to the action of performing payment	B, M, D	planning	planning
the joint commitment to nominal action has been dropped	B, M, D	agent reasoning	n/a

Table 9 continued: trade team beliefs concerning teamwork

Belief	Agent	Addition Triggers	Drop Triggers
an agent has dropped its commitment to nominal action	B, M, D	agent reasoning	n/a
there is a mutual belief that the joint commitment to nominal action has been dropped	B, M, D	reasoning about self beliefs and reasoning about communication	n/a
trade team is in the planning state	B, M, D	a joint commitment to nominal action has been reached	n/a
merchant m has committed (dropped commitment) to the action of receiving payment	B, M, D	planning	planning
delivery d has committed (dropped commitment) to the action of delivering merchandise to buyer b	B, M, D	planning	planning
merchant m has committed (dropped commitment) to finalizing its part of the trade transaction	B, M, D	planning	planning

Table 9 continued: trade team beliefs concerning teamwork

Belief	Agent	Addition Triggers	Drop Triggers
buyer b has committed (dropped commitment) to the action of receiving merchandise from delivery d	B, M, D	planning	planning
buyer b has committed (dropped commitment) to finalizing its part of the trade transaction	B, M, D	planning	planning
buyer b has committed (dropped commitment) to the joint action of performing trade	B, M, D	planning	planning
merchant m has committed (dropped commitment) to the joint action of performing trade	B, M, D	planning	planning
delivery d has committed (dropped commitment) to the joint action of performing trade	B, M, D	planning	planning

Table 9 continued: trade team beliefs concerning teamwork

Belief	Agent	Addition Triggers	Drop Triggers

a joint commitment to the joint action of performing trade has (not) been reached	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication
there is a mutual belief that a there is a joint commitment to joint action of performing trade has (not) been reached	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication
trade team is in the post-planning state	B, M, D	a joint commitment to the joint action of performing trade has been reached	a joint commitment to the joint action of performing trade has been dropped
buyer b has intended the joint action of performing trade	B, M, D	agent reasoning about task execution	agent reasoning about task execution
merchant m has intended the joint action of performing trade	B, M, D	agent reasoning about task execution	agent reasoning about task execution
there is a mutual belief that a there is a joint intention to joint action of performing trade has (not) been reached	B, M, D	agent reasoning about task execution and agent communication	agent reasoning about task execution and agent communication

Table 9 continued: trade team beliefs concerning teamwork

Belief	Agent	Addition Triggers	Drop Triggers
delivery d has	B, M,	agent reasoning about	agent reasoning about task

intended the joint action of performing trade	D	task execution	execution
a joint intention to the joint action of performing trade has (not) been reached	B, M, D	agent reasoning about task execution and agent communication	agent reasoning about task execution and agent communication
trade team is in the execution state	B, M, D	joint intention of joint action of performing trade has been reached	joint intention of joint action of performing trade has been dropped
merchant m is executing the task of committing its part of trade transaction	B, M, D	reasoning about task execution	reasoning about task execution
buyer b is executing the task of receiving merchandise	B, M, D	reasoning about task execution	reasoning about task execution
buyer b is executing the task of committing its part of trade transaction	B, M, D	reasoning about task execution	reasoning about task execution
merchant m is executing the task of receiving payment	B, M, D	reasoning about task execution	reasoning about task execution

Table 9 continued: trade team beliefs concerning teamwork

Belief	Agent	Addition Triggers	Drop Triggers
trade team is	B, M,	reasoning about task	reasoning about task

executing the task of performing trade	D	execution	execution
buyer b is executing the task of performing payment	B, M, D	reasoning about task execution	reasoning about task execution
buyer b has achieved the goal of committing its part of trade transaction	B, M, D	reasoning about task execution	reasoning about task execution
the goal of performing payment is achieved, unachieved, or irrelevant	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication
the goal of receiving payment is achieved, unachieved, or irrelevant	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication
the goal of delivering merchandise is achieved, unachieved, or irrelevant	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication

Table 9 continued: trade team beliefs concerning teamwork

Belief	Agent	Addition Triggers	Drop Triggers
the goal of receiving	B, M,	agent reasoning about	agent reasoning about self

merchandise is achieved, unachieved, or irrelevant	D	self and about communication	and about communication
the goal of committing merchant part of trade is achieved, unachieved, or irrelevant	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication
team execution has ended	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication
the goal of committing buyer part of trade is achieved, unachieved, or irrelevant	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication
the goal of performing trade is achieved, unachieved, or irrelevant	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication

Table 9 continued: trade team beliefs concerning teamwork

Belief	Agent	Addition Triggers	Drop Triggers

it is mutually believed within the team that the goal of performing trade is achieved, unachieved, unachievable, or irrelevant	B, M, D	agent reasoning about self and about communication	agent reasoning about self and about communication

Table 9 continued: trade team beliefs concerning teamwork

5.5 Trade Team Collaborative Interactions

Within a team of trade agents, team agents contribute to a number of collaborative interactions that can be classified as follows:

Team formation interaction: Team formation interaction execute with the objective of forming a team to perform purchase transaction with a joint commitment to nominal action that is believed to achieve that goal.

Planning interaction: Planning interaction executes with the goal of jointly building a hierarchy of commitments to actions that is believed to achieve the trade team goal of performing the purchase. The hierarchy is built taken into consideration goal, commitment, and intention dependencies.

Team execution interaction: Team execution interaction execute as team agents collectively cooperate to achieve designated goal, executing joint and individual actions. As agents execute actions, they jointly build a hierarchy of intentions.

5.5.1 Executing Collaboration Scenarios: An Example

Next, we introduce the mechanism used by trade agents to execute collaboration scenarios. We start by introducing agent plans and their relationships. Next, we introduce plans used by the buyer agent to form a trade team.

5.5.1.1 Team Formation Plan Structure

The team formation plan structure consists of one main plan, the form_team plan, and three sub-plans, the attempt_to_solicit_assistance plan, the verify_mutual_belief plan and the verify_joint_commitment_to_nominal_action plan. Next, we shall look at each of these plans.

Form_Team Plan

Description:

The buyer agent uses this plan in order to construct a trade team consisting of a buyer, merchant and delivery agents. The buyer executes a set of steps that follow the form team conversational pattern described in chapter 3. The plan succeeds if its post conditions were satisfied.

Pre-conditions:

- The belief that team is in the pre-planning team state
- The belief that trade is not performed yet
- The belief that the buyer should perform trade
- The belief that the buyer b, merchant m and delivery d are able to jointly perform trade
- The perform trade persistent goal exists

Post-conditions:

- The belief that a mutual belief regarding team's ability has (not) been reached
- The belief that a joint commitment to nominal action has (not) been reached
- In case of team formation success, the addition of a commitment to nominal action
- The belief that buyer agent has succeeded (failed) in its attempt to form a team
- The belief that the current team state is (pre-)planning team state

Actions:

1. adopt the belief that the buyer should solicit assistance from merchant m and delivery d
2. Announce buyer's belief that the trade team is able to perform trade
3. Commit to nominal action relative to the goal of performing trade
4. Announce commitment to nominal actions to merchant m and delivery d.
5. if the verify_mutual_belief plan succeeds, update team state to the planning state

Attempt_To_Solicit_Assistance Plan

Description:

Buyer attempts to solicit assistance from merchant m and delivery d. This plan is a sub-plan for the form_team plan

Pre-conditions:

- the belief that the buyer should solicit assistance from merchant m and delivery d)

Post-conditions:

- the buyer agent has added a commitment to nominal action

Actions:

1. Announce attempt to solicit assistance to merchant m and delivery d.

Verify_Mutual_Belief Plan

Description:

This plan is used to verify the status of mutual belief regarding team's ability to achieve the perform trade goal. This plan is a sub-plan for the form_team plan

Pre-conditions:

- Announce buyer belief that the trade team is able to perform trade

Post-conditions:

- The belief that a mutual belief regarding team ability to achieve goal has (not) been reached.

Actions:

1. query individual beliefs of merchant m and delivery d regarding team ability to achieve team
2. based on their replies, verify the state of mutual belief

Verify_Joint_Commitment_To_Nominal_Action Plan

Description:

This plan is used to verify the status of commitment to team's nominal action. This plan is a sub-plan for the form_team plan

Pre-conditions:

- Buyer agent has committed to nominal action relative to achieving the perform trade goal.

Post-conditions:

- The belief about the status of joint commitment to nominal action relative to achieving the perform trade goal has (not) been reached.

Actions:

1. query individual beliefs of merchant m and delivery d regarding team ability to achieve team
2. based on their replies, verify the state of joint commitment

5.5.1.2 Executing the Form Team Scenarios by the Buyer Agent

The selection of plans for execution is performed based on the plan pre-conditions set. Next, we describe a single scenario for the execution of buyer agent's form_team plan and its sub-plans.

1. At first, the form_team is executed as its pre-conditions match required criteria.
2. The first action step in the form_team plan executes, leading to the firing of the attempt_to_solicit_assistance sub-plan as its pre-conditions are matched.
3. The first step of the attempt_to_solicit_assistance sub-plan executes, and the agent announces to the merchant and delivery agents its attempt to solicit assistance. Next, the sub-plan ends execution successfully and the control returns to the parent plan.
4. The second step of the form_team plan executes, and the control is then transformed to the verify_mutual_belief sub-plan since its pre-conditions are matched.
5. The first step in the verify_mutual_belief sub-plan is executed.
6. The second step of the verify_mutual_belief sub-plan is executed, the sub-plan execution ends successfully and the control returns to the form_team plan.
7. The third and fourth steps of the form_team plan execute, and the control is transformed to the verify_joint_commitment_to_nominal_action sub-plan as its pre-conditions is matched.
8. The first step of the verify_joint_commitment_to_nominal_action sub-plan is executed.
9. The second step of the verify_joint_commitment_to_nominal_action sub-plan is executed successfully and the control is transformed to the form_team plan.
10. The fifth step of the form_team plan is executed, verifying the success status of the verify_joint_commitment_to_nominal_action sub-plan plan, then the execution of the form_team plan ends successfully.

5.6 Collaborative Behavior Analysis Facility

In this section, we describe the development of a software program developed to support the development of collaborating trade agents.

5.6.1 An Approach for the Specification of MAS Collaborative Behavior

In order to specify collaborative behavior for a trade MAS, it is required to define how agents should behave in different collaborative situations. For example, what should happen if the delivery agent dropped its commitment to the trade joint action. An answer to such question depends on the strategy adopted. One approach is for both the merchant and buyer agents to drop their commitments, another approach is to try to convince the delivery agent before deciding to drop their commitment. A specification can define the trade system's collaborative behavior, which MAS developers would then use in developing and testing MAS agents.

The specification trade MAS collective behavior should enable the definition of the following components:

- interaction scenarios
- concrete definition of the shared and individual mental states of a collective with associated deliberation rules
- the conversations exchanged between agents.

Changing the collaborative strategy would cause the specification to change, affecting interaction scenarios, the definition of shared mental state with its associated deliberation rules, and hence, the conversations used. What is required is a flexible approach to define the specification that would enable changes to be specified, developed, and tested in an organized manner.

Our proposed approach to the specification of collaborative behavior is through using rapid prototyping to develop a program that imitates the collaborative behavior of real trade agents using a given collaborative strategy. The program input is a set of instructions adopted by each simulating agent as its set of beliefs about collaborative behavior actions. Testing how the agent should behalf is performed through the generation of a log of interaction messages. By defining all valid collaborative situations and validating team agents' behavior at these different situations, we would have defined the specification of MAS collaborative behavior. The specification product consists of input scenarios, the prototype itself, and the output behavior. The specification is used as to develop and validate MAS system's collaborative behavior. Figure 12 outlines the proposed approach.

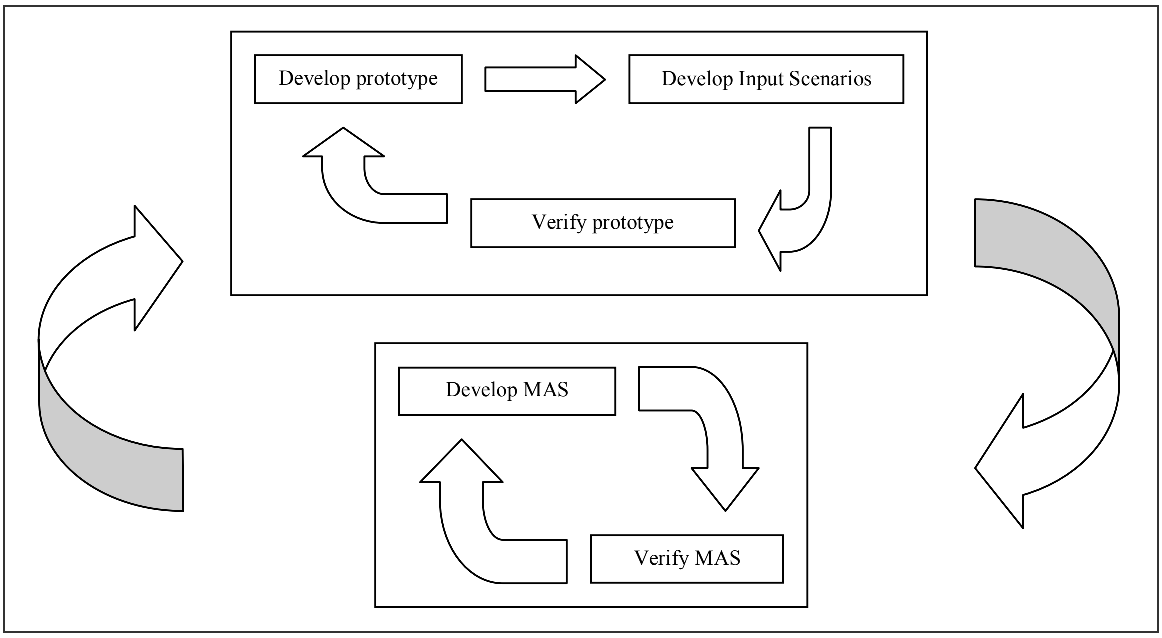

Figure 12: A proposed iterative approach for the rapid prototyping of MAS cooperative behavior

5.6.2 Program Input

The program takes as input a command file containing a list of names of input file to be processed. Each input file contains a list of behavioral instructions. Each behavioral instruction has some setting that the program uses later in the program. Trade agent's role categorizes behavioral instructions. For example, a behavioral instruction of committing to perform payment corresponds to the role of a buyer. An input file contains behavioral instructions for each of the trade agents.

5.6.3 Program Processing

The program reads input files one at a time in sequence and for each file, processes it, and generates a separate text file containing generated interaction messages. Each of the simulated trade agents loads its set of behavioral instructions for later use.

The program simulates the four phases of CPS. Agents reason about team state and contribute to the following interaction scenarios:

1. form a trade team by having the buyer soliciting assistance from the merchant and the delivery agents
2. build an action commitment hierarchy, and
3. simulate trade action execution, building a hierarchy of intentions

At each of these phases, agents maintain beliefs about other agents' commitments, intentions, and action execution status. In addition, loaded behavioral instructions acts as being agent's beliefs about what to do at a given deliberative situations. For example, a behavioral instruction may direct an agent not to commit to a given designated action. This instruction is retained as an agent belief that an agent would use at deliberation time to decide whether to commit to that action or not.

On the other hand, the program implements the following strategy:

- If an agent drops a commitment to execute an action, then the agent on which the dropped action is dependent should drop its commitment
- If the perform payment action is dropped, then agent should drop joint commitment to main action
- If an agent's intention to the joint action is dropped, then other agents should drop their intentions to that action
- If an agent drops its intention to execute an action, then the trade behavior would freeze
- If an action attempt fails, then the agent would drop its intention to execute that action

5.6.4 Program Output

The program generates a log file containing teamwork messages exchanged between simulated agents in a chronological order. The messages are interleaved with simulated agent traces containing text message describing simulated agent reasoning about team collective behavior.

5.6.5 Sample Data

Appendix A contains a sample input file for the best case where everything goes without problems. Appendix B contains the output log as generated by processing the input file. In appendix A, a textual description for each team message is embedded.

5.6.6 The Use of the Framework Implementation

The program makes use of the proposed framework specification the framework proposed implementation, generating a framework compatible XML messages supporting ACL and BRL. This enables the validation of MAS agents generated interaction messages.

5.7 The Case Study Findings

The case study findings can be summarized as follows:

a. Agent interaction and communication is crucial for maintaining a shared and consistent view of the trade problem. Agents were required to express their mental states in a clear and consistent manner understandable by other agents. Interaction enabled agents to reason about coordination and cooperation.

b. A common view of the goals, actions, commitments, and intentions, were required since this view helped agents to make their own decision regarding teamwork activities and state.

c. The use of framework conversational model helped agents contribute interaction scenarios required to develop and maintain team's mutual understanding and team's shared view of the trade. In addition, it enabled agents to consistently reason about team states.

d. The framework implementation enabled agents express collaborative mental behavior, using a set of agent interaction mechanisms, and transmitted using a message interchange format.

e. By reviewing application program log, the agent developer is able to verify team and individual collaborative behavior.

Final Conclusion

Multi-agent systems are complex systems consisting of a number of agents, each of which by itself might represent an organization consisting of one or more agents. As part of a multi-agent system, an agent has a degree of autonomy in making its own decisions. Agents interact within a system of agents in order to achieve their individual and collective goals in an act defined as collaboration. Collaboration occurs when a group of agents are acting together, developing and maintaining a shared team mental state.

Current MAS environments supporting multi-agent collaboration lack the characteristics of open environments. On the other hand, open environments lack the existence of a general collaboration framework required to supports the development and maintenance of shared team mental state among heterogeneous team agents. Existing multi-agent standards, development tools, and environments support the development of MAS through providing support for communication, planning, coordination, cooperation, task allocation and task execution activities. On the hand, the proposed framework does not assume the existence of a fixed set of interaction protocols but rather. In addition, the proposed framework is transparent to existing tools since there is no overlap in functionality between them.

The framework design maintains a clear separation between the framework specification and the possible framework implementations, enabling the creation of implementations that match the configurations of open environment.

The proposed framework implementation provided a behavior representation language that supports modal and temporal constructs, enabling agents to reason about the possibility and time. In addition, it provided an agent communication language that allows agent to express and query other agents' mental state. The proposed framework implementation also assumes a layered conversational model, defines a message interchange format for representing team messages in XML, and finally, the proposed framework implementation provides a XML message encoding/decoding facility for encoding and decoding team messages.

The use of collaborative agents would enhance e-commerce systems, enhancing flexibility and dynamism for such heterogeneous systems, for example, in the trade domain, if the payment fails due to balance of one account, an agent might individually try to use another account for payment, and if it fails, it might re-negotiate the payment method or payment terms with the merchant agent.

The trade purchase problem can be modeled as a number of trading entities acting together to select, negotiate and execute trade transactions. The framework specification and implementation enabled agents to collaborate to execute trade.

Finally, the iterative proposed approach for prototyping trade team agents' collaborative behavior enables the rapid development and maintenance of the specification of MAS collaborative behavior.

Appendix A:
A Sample Input File for the Analysis Facility

Following is a sample input file to the collaborative behavior analysis tool. Each line represents an instruction that determines the program behavior at a given decision point. For example, the `buyer.CommitToNominalAction=YES` instruction is used by the buyer agent to decide to commit to a nominal action relative to the achievement of main goal. Input instructions are related to the decisions that agents take during the execution of collaborative conversations.

```
buyer.CommitToNominalAction=YES
buyer.CommitToPerformTrade=YES
buyer.CommitToPerformTrade=YES
buyer.IntendPerformTrade=YES
buyer.CommitToPerformPayment=YES
buyer.CommitToReceiveMerchandise=YES
buyer.CommitToBuyerCommitTrans=YES
buyer.IntendPerformPayment=YES
buyer.IntendReceiveMerchandise=YES
buyer.IntendBuyerCommitTrans=YES
buyer.AttemptPerformPayment=YES
buyer.AttemptReceiveMerchandise=YES
buyer.AttemptBuyerCommitTrans=YES
merchant.CommitToNominalAction=YES
merchant.CommitToPerformTrade=YES
merchant.CommitToPerformTrade=YES
merchant.IntendPerformTrade=YES
merchant.CommitToReceivePayment=YES
merchant.CommitToMerchantCommitTrans=YES
merchant.IntendReceivePayment=YES
merchant.IntendMerchantCommitTrans=YES
merchant.AttemptReceivePayment=YES
merchant.AttemptMerchantCommitTrans=no
delivery.CommitToNominalAction=YES
delivery.CommitToPerformTrade=YES
delivery.CommitToPerformTrade=YES
delivery.IntendPerformTrade=YES
delivery.CommitToDeliverMerchandise=YES
delivery.IntendDeliverMerchandise=YES
delivery.AttemptDeliverMerchandise=YES
```

Appendix B:
A Sample Interaction for Performing e-Trade

Below is a log generated by the collaborative behavior analysis tool for the input file described in Appendix A. XML messages are annotated by textual description for the message and its sending agent. The log instance represents a successful trade execution by the trade team.

At startup, trade agents individually reason that the current team state is the 'pre-planning'.
buyer: updating team state to 'pre-planning' merchant: updating team state to 'pre-planning' delivery: updating team state to 'pre-planning'
The buyer agent announces its attempt to solicit assistance


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_1">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info
          sender_id="buyer">
          <recipient_list>
               <recipient agent_id="merchant" />
               <recipient agent_id="delivery" />
          </recipient_list>
     <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
                    bdi_clause_id =
"SolicitAssistance_attempt"
               type="added" id="SolicitAssistance" />
          <bdi_clause
               entity_id="buyer"
               type="goal"
               id="SolicitAssistance_goal" />
          <bdi_clause
               action_id="SolicitAssistance_action"
                    entity_id="buyer"
```

```
     commitment_id="SolicitAssistance_commitment"
          type="intention"
               id="SolicitAssistance_intention" />
          <bdi_clause
               entity_id="buyer"
               type="action"
               id="SolicitAssistance_action" />
          <bdi_clause
               entity_id="buyer"
               type="attempt"

     intention_id="SolicitAssistance_intention"
          id="SolicitAssistance_attempt" />
          <bdi_clause
               entity_id="buyer"
                    goal_id="SolicitAssistance_goal"
               type="commitment"
                    id="SolicitAssistance_commitment" />
     </message_content>
</team_message>
```

The buyer agent announces its belief that the goal of performing trade is achievable

```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_2">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="buyer">
          <recipient_list>
               <recipient agent_id="merchant" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
                    bdi_clause_id="perform_trade_goal"
                         type="can_achieve"
               id="perform_trade" />
          <bdi_clause
               entity_id="buyer"
               type="goal"
               id="perform_trade_goal" />
     </message_content>
</team_message>
```

The buyer agent queries merchant and delivery agents beliefs regarding the achievability of the perform trade goal


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_3">
      <meta_info
            content_type="framework-implementation" />
      <delivery_info sender_id="buyer">
            <recipient_list>
                  <recipient agent_id="merchant" />
            </recipient_list>
            <reply_to_list>
                  <recipient agent_id="buyer" />
                  <recipient agent_id="delivery" />
            </reply_to_list>
      </delivery_info>
      <message_content>
            <message_clause
                  truth_value="TeamAbility"
                        bdi_clause_id="perform_trade_goal"
                  type="can_achieve"
                  id="perform_trade" />
            <bdi_clause
                  entity_id="merchant"
                  type="goal"
                  id="perform_trade_goal" />
      </message_content>
</team_message>

<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_4">
      <meta_info
            content_type="framework-implementation" />
      <delivery_info sender_id="buyer">
            <recipient_list>
                  <recipient agent_id="delivery" />
            </recipient_list>
            <reply_to_list>
                  <recipient agent_id="buyer" />
                  <recipient agent_id="merchant" />
            </reply_to_list>
      </delivery_info>
      <message_content>
            <message_clause
                  truth_value="TeamAbility"
                        bdi_clause_id="perform_trade_goal"
                             type="can_achieve"
```

```
                id="perform_trade" />
          <bdi_clause
                entity_id="merchant"
                type="goal"
                id="perform_trade_goal" />
     &lt;/message_content&gt;
&lt;/team_message&gt;
```

The delivery agent replies to buyer's query, announcing its belief that the perform trade goal is achievable

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
<team_message
     team_id="trade_team"
     message_id="MESSAGE_5">
     <meta_info
          content_type="framework-implementation" />
     &lt;delivery_info sender_id="delivery"&gt;
          &lt;recipient_list&gt;
               &lt;recipient agent_id="buyer" /&gt;
               &lt;recipient agent_id="merchant" /&gt;
          &lt;/recipient_list&gt;
          &lt;reply_to_list&gt;&lt;/reply_to_list&gt;
     &lt;/delivery_info&gt;
     &lt;message_content&gt;
          <message_clause
               truth_value="true"
                    bdi_clause_id="perform_trade_goal"
                         type="can_achieve"

     id="can_achieve_perform_trade" />
          <bdi_clause
               entity_id="delivery"
               type="goal"
               id="perform_trade_goal" />
     &lt;/message_content&gt;
&lt;/team_message&gt;
```

The merchant agent replies to buyer's query, announcing its belief that the perform trade goal is achievable

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
<team_message
     team_id="trade_team"
     message_id="MESSAGE_6">
     <meta_info
          content_type="framework-implementation" />
     &lt;delivery_info sender_id="merchant"&gt;
```

```
          <recipient_list>
               <recipient agent_id="buyer" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
                    bdi_clause_id="perform_trade_goal"
               type="can_achieve"
               id="can_achieve_perform_trade" />
          <bdi_clause
               entity_id="merchant"
               type="goal"
               id="perform_trade_goal" />
     </message_content>
</team_message>
```


The buyer agent announces its commitment to nominal action in order to achieve the perform trade goal


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_7">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="buyer">
          <recipient_list>
               <recipient agent_id="merchant" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
               bdi_clause_id="NominalAction_commitment"
               type="added" id="NominalAction" />
          <bdi_clause
               entity_id="buyer"
               type="action"
               id="NominalAction_action" />
          <bdi_clause
               entity_id="buyer"
               type="goal" id="NominalAction_goal" />
          <bdi_clause
               entity_id="buyer"
               goal_id="NominalAction_goal"
               type="commitment"
```

```
            id="NominalAction_commitment" />
    </message_content>
</team_message>
```


The buyer agent queries the merchant and delivery agents regarding their commitment to nominal action in order to achieve the perform trade goal


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
    team_id="trade_team"
    message_id="MESSAGE_8">
    <meta_info
        content_type="framework-implementation" />
    <delivery_info sender_id="buyer">
        <recipient_list>
            <recipient agent_id="merchant" />
        </recipient_list>
        <reply_to_list>
            <recipient agent_id="buyer" />
            <recipient agent_id="delivery" />
        </reply_to_list>
    </delivery_info>
    <message_content>
        <message_clause
            truth_value="query_commitment"
            bdi_clause_id="NominalAction_commitment"
            type="added" id="NominalAction" />
        <bdi_clause
            entity_id="buyer"
            type="action"
            id="NominalAction_action" />
        <bdi_clause
            entity_id="buyer"
            type="goal"
            id="NominalAction_goal" />
        <bdi_clause
            entity_id="buyer"
            goal_id="NominalAction_goal"
            type="commitment"
            id="NominalAction_commitment" />
    </message_content>
</team_message>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
    team_id="trade_team"
    message_id="MESSAGE_9">
    <meta_info
        content_type="framework-implementation" />
    <delivery_info sender_id="buyer">
```

```
          <recipient_list>
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list>
               <recipient agent_id="buyer" />
               <recipient agent_id="delivery" />
          </reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="query_commitment"
               bdi_clause_id="NominalAction_commitment"
               type="added"
               id="NominalAction" />
          <bdi_clause
               entity_id="buyer"
               type="action"
               id="NominalAction_action" />
          <bdi_clause
               entity_id="buyer"
               type="goal"
               id="NominalAction_goal" />
          <bdi_clause
               entity_id="buyer"
               goal_id="NominalAction_goal"
               type="commitment"
               id="NominalAction_commitment" />
     </message_content>
</team_message>
```

The delivery agent announces its commitment to nominal action in order to achieve the perform trade goal

```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_10">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="delivery">
          <recipient_list>
               <recipient agent_id="buyer" />
               <recipient agent_id="merchant" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
               bdi_clause_id="NominalAction_commitment"
               type="added" id="NominalAction" />
```

```
            <bdi_clause
                entity_id="delivery"
                type="action"
                id="NominalAction_action" />
            <bdi_clause
                entity_id="delivery"
                type="goal"
                id="NominalAction_goal" />
            <bdi_clause
                entity_id="delivery"
                goal_id="NominalAction_goal"
                type="commitment"
                id="NominalAction_commitment" />
        </message_content>
</team_message>
```


The merchant agent announces its commitment to nominal action in order to achieve the perform trade goal


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
        team_id="trade_team"
        message_id="MESSAGE_11">
        <meta_info
            content_type="framework-implementation" />
        <delivery_info sender_id="merchant">
            <recipient_list>
                <recipient agent_id="buyer" />
                <recipient agent_id="delivery" />
            </recipient_list>
            <reply_to_list></reply_to_list>
        </delivery_info>
        <message_content>
            <message_clause
                truth_value="true"
                bdi_clause_id="NominalAction_commitment"
                type="added"
                id="NominalAction" />
            <bdi_clause
                entity_id="merchant"
                type="action"
                id="NominalAction_action" />
            <bdi_clause
                entity_id="merchant"
                type="goal"
                id="NominalAction_goal" />
            <bdi_clause
                entity_id="merchant"
                goal_id="NominalAction_goal"
                type="commitment"
                id="NominalAction_commitment" />
```

```
     </message_content>
</team_message>
```

Trade agents individually reason a joint commitment to nominal action has been reached, and hence, updating their agent's team state into the team planning state

```
buyer: updating team state to 'planning'
merchant: updating team state to 'planning'
delivery: updating team state to 'planning'
```

The buyer agent announces its commitment to the perform payment action, relative to the main goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_12">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="buyer">
          <recipient_list>
               <recipient agent_id="merchant" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
               bdi_clause_id="PerformPayment_commitment"
               type="added"
               id="PerformPayment" />
          <bdi_clause
               entity_id="buyer"
               goal_id="PerformPayment_goal"
               type="commitment"
id="PerformPayment_commitment" />
          <bdi_clause
               entity_id="buyer"
               type="goal"
               id="PerformPayment_goal" />
          <bdi_clause
               entity_id="buyer"
               type="action"
               id="PerformPayment_action" />
     </message_content>
</team_message>
```

The merchant agent announces its commitment to the receive payment action, relative to the main goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_13">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="merchant">
          <recipient_list>
               <recipient agent_id="buyer" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
               bdi_clause_id="ReceivePayment_commitment"
               type="added"
               id="ReceivePayment" />
          <bdi_clause
               entity_id="merchant"
               type="action"
               id="ReceivePayment_action" />
          <bdi_clause
               entity_id="merchant"
               type="goal"
               id="ReceivePayment_goal" />
          <bdi_clause
               entity_id="merchant"
               goal_id="ReceivePayment_goal"
               type="commitment"
               id="ReceivePayment_commitment" />
     </message_content>
</team_message>
```


The delivery agent announces its commitment to the deliver merchandise action, relative to the main goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_14">
     <meta_info
          content_type="framework-implementation" />
```

```
      <delivery_info sender_id="delivery">
           <recipient_list>
                <recipient agent_id="buyer" />
                <recipient agent_id="merchant" />
           </recipient_list>
           <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
           <message_clause
                truth_value="true"
                bdi_clause_id=
                          "DeliverMerchandise_commitment"
                type="added"
                id="DeliverMerchandise" />
           <bdi_clause
                entity_id="delivery"
                goal_id="DeliverMerchandise_goal"
                type="commitment"
                id="DeliverMerchandise_commitment" />
           <bdi_clause
                entity_id="delivery"
                type="goal"
                id="DeliverMerchandise_goal" />
           <bdi_clause
                entity_id="delivery"
                type="action"
                id="DeliverMerchandise_action" />
      </message_content>
</team_message>
```

The buyer agent announces its commitment to the receive merchandise action, relative to the main goal of performing trade

```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_15">
      <meta_info
           content_type="framework-implementation" />
      <delivery_info sender_id="buyer">
           <recipient_list>
                <recipient agent_id="merchant" />
                <recipient agent_id="delivery" />
           </recipient_list>
           <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
           <message_clause
                truth_value="true"
```

```
                bdi_clause_id =

      "ReceiveMerchandise_commitment"
                type="added" id="ReceiveMerchandise" />
           <bdi_clause
                entity_id="buyer"
                goal_id="ReceiveMerchandise_goal"
                type="commitment"
                id="ReceiveMerchandise_commitment" />
           <bdi_clause
                entity_id="buyer"
                type="action"
                id="ReceiveMerchandise_action" />
           <bdi_clause
                entity_id="buyer"
                type="goal"
                id="ReceiveMerchandise_goal" />
      </message_content>
</team_message>
```


The merchant agent announces its commitment to the settlement of the merchant's part of the transaction action, relative to the main goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_16">
      <meta_info
           content_type="framework-implementation" />
      <delivery_info sender_id="merchant">
           <recipient_list>
                <recipient agent_id="buyer" />
                <recipient agent_id="delivery" />
           </recipient_list>
           <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
           <message_clause
                truth_value="true"
                bdi_clause_id =
                          "MerchantCommitTrans_commitment"
                type="added"
                id="MerchantCommitTrans" />
           <bdi_clause
                entity_id="merchant"
                type="action"
                id="MerchantCommitTrans_action" />
           <bdi_clause
                entity_id="merchant"
```

```
                type="goal"
                id="MerchantCommitTrans_goal" />
           <bdi_clause
                entity_id="merchant"
                goal_id="MerchantCommitTrans_goal"
                type="commitment"
                id="MerchantCommitTrans_commitment" />
      </message_content>
</team_message>
```


The buyer agent announces its commitment to the settlement of the buyer's part of the transaction action, relative to the main goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_17">
      <meta_info
           content_type="framework-implementation" />
      <delivery_info sender_id="buyer">
           <recipient_list>
                <recipient agent_id="merchant" />
                <recipient agent_id="delivery" />
           </recipient_list>
           <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
           <message_clause
                truth_value="true"

      bdi_clause_id="BuyerCommitTrans_commitment"
                type="added" id="BuyerCommitTrans" />
           <bdi_clause
                entity_id="buyer"
                goal_id="BuyerCommitTrans_goal"
                type="commitment"
                id="BuyerCommitTrans_commitment" />
           <bdi_clause
                entity_id="buyer"
                type="action"
                id="BuyerCommitTrans_action" />
           <bdi_clause
                entity_id="buyer"
                type="goal"
                id="BuyerCommitTrans_goal" />
      </message_content>
</team_message>
```

The buyer agent announces its commitment to the collective action of performing trade, relative to the main goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_18">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="buyer">
          <recipient_list>
               <recipient agent_id="merchant" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
               bdi_clause_id="PerformTrade_commitment"
               type="added"
               id="PerformTrade" />
          <bdi_clause
               entity_id="buyer"
               type="goal"
               id="PerformTrade_goal" />
          <bdi_clause
               entity_id="buyer"
               goal_id="PerformTrade_goal"
               type="commitment"
               id="PerformTrade_commitment" />
          <bdi_clause
               entity_id="buyer"
               type="action"
               id="PerformTrade_action" />
     </message_content>
</team_message>
```


The merchant agent announces its commitment to the collective action of performing trade, relative to the main goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_19">
     <meta_info
          content_type="framework-implementation" />
```

```
      <delivery_info sender_id="merchant">
           <recipient_list>
                <recipient agent_id="buyer" />
                <recipient agent_id="delivery" />
           </recipient_list>
           <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
           <message_clause
                truth_value="true"
                bdi_clause_id="PerformTrade_commitment"
                type="added"
                id="PerformTrade" />
           <bdi_clause
                entity_id="merchant"
                type="goal"
                id="PerformTrade_goal" />
           <bdi_clause
                entity_id="merchant"
                goal_id="PerformTrade_goal"
                type="commitment"
                id="PerformTrade_commitment" />
           <bdi_clause
                entity_id="merchant"
                type="action"
                id="PerformTrade_action" />
      </message_content>
</team_message>
```


The delivery agent announces its commitment to the collective action of performing trade, , relative to the main goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_20">
      <meta_info
           content_type="framework-implementation" />
      <delivery_info sender_id="delivery">
           <recipient_list>
                <recipient agent_id="buyer" />
                <recipient agent_id="merchant" />
           </recipient_list>
           <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
           <message_clause
                truth_value="true"
                bdi_clause_id="PerformTrade_commitment"
```

```
                type="added"
                id="PerformTrade" />
           <bdi_clause
                entity_id="delivery"
                type="goal"
                id="PerformTrade_goal" />
           <bdi_clause
                entity_id="delivery"
                goal_id="PerformTrade_goal"
                type="commitment"
                id="PerformTrade_commitment" />
           <bdi_clause
                entity_id="delivery"
                type="action"
                id="PerformTrade_action" />
      </message_content>
</team_message>
```


Trade agents individually reason that a joint commitment to the perform trade action has been reached and hence, trade agents reason that the team state has changed to the 'post-planning' team state

buyer: updating team state to 'post-planning'
merchant: updating team state to 'post-planning'
delivery: updating team state to 'post-planning'

The delivery agent intends the main action of performing trade relative to the goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_21">
      <meta_info
           content_type="framework-implementation" />
      <delivery_info sender_id="delivery">
           <recipient_list>
                <recipient agent_id="buyer" />
                <recipient agent_id="merchant" />
           </recipient_list>
           <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
           <message_clause
                truth_value="true"
                bdi_clause_id="PerformTrade_intention"
                type="added"
```

```
                id="PerformTrade" />
          <bdi_clause
                action_id="PerformTrade_action"
                entity_id="delivery"
                commitment_id="PerformTrade_commitment"
                type="intention"
                id="PerformTrade_intention" />
          <bdi_clause
                entity_id="delivery"
                type="goal"
                id="PerformTrade_goal" />
          <bdi_clause
                entity_id="delivery"
                goal_id="PerformTrade_goal"
                type="commitment"
                id="PerformTrade_commitment" />
          <bdi_clause
                entity_id="delivery"
                type="action"
                id="PerformTrade_action" />
     &lt;/message_content&gt;
&lt;/team_message&gt;
```

The buyer agent intends the main action of performing trade relative to the goal of performing trade

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
<team_message
     team_id="trade_team"
     message_id="MESSAGE_22">
     <meta_info
          content_type="framework-implementation" />
     &lt;delivery_info sender_id="buyer"&gt;
          &lt;recipient_list&gt;
                &lt;recipient agent_id="merchant" /&gt;
                &lt;recipient agent_id="delivery" /&gt;
          &lt;/recipient_list&gt;
          &lt;reply_to_list&gt;&lt;/reply_to_list&gt;
     &lt;/delivery_info&gt;
     &lt;message_content&gt;
          <message_clause
                truth_value="true"
                bdi_clause_id="PerformTrade_intention"
                type="added"
                id="PerformTrade" />
          <bdi_clause
                action_id="PerformTrade_action"
                entity_id="buyer"
                commitment_id="PerformTrade_commitment"
                type="intention"
                id="PerformTrade_intention" />
```

```
          <bdi_clause
               entity_id="buyer"
               type="goal"
               id="PerformTrade_goal" />
          <bdi_clause
               entity_id="buyer"
               goal_id="PerformTrade_goal"
               type="commitment"
               id="PerformTrade_commitment" />
          <bdi_clause
               entity_id="buyer"
               type="action"
               id="PerformTrade_action" />
     </message_content>
</team_message>
```


The merchant agent intends the main action of performing trade relative to the goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_23">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="merchant">
          <recipient_list>
               <recipient agent_id="buyer" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
               bdi_clause_id="PerformTrade_intention"
               type="added"
               id="PerformTrade" />
          <bdi_clause
               action_id="PerformTrade_action"
               entity_id="merchant"
               commitment_id="PerformTrade_commitment"
               type="intention"
               id="PerformTrade_intention" />
          <bdi_clause
               entity_id="merchant"
               type="goal"
               id="PerformTrade_goal" />
          <bdi_clause
               entity_id="merchant"
               goal_id="PerformTrade_goal"
```

```
                type="commitment"
                id="PerformTrade_commitment" />
        <bdi_clause
                entity_id="merchant"
                type="action"
                id="PerformTrade_action" />
    </message_content>
</team_message>
```


Trade agents individually reason that a joint intention to the main action has been reached and hence, trade agents reason that team state has changed to the execution state

buyer: updating team state to 'execution'
merchant: updating team state to 'execution'
delivery: updating team state to 'execution'

The buyer agent intends the perform payment action as part of the joint intention to the goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
    team_id="trade_team"
    message_id="MESSAGE_24">
    <meta_info
        content_type="framework-implementation" />
    <delivery_info sender_id="buyer">
        <recipient_list>
            <recipient agent_id="merchant" />
            <recipient agent_id="delivery" />
        </recipient_list>
        <reply_to_list></reply_to_list>
    </delivery_info>
    <message_content>
        <message_clause
            truth_value="true"
            bdi_clause_id="PerformPayment_intention"
            type="added"
            id="PerformPayment" />
        <bdi_clause
            action_id="PerformPayment_action"
            entity_id="buyer"
            commitment_id="PerformPayment_commitment"
            type="intention"
            id="PerformPayment_intention" />
        <bdi_clause
            entity_id="buyer"
            goal_id="PerformPayment_goal"
```

```
                type="commitment"
                id="PerformPayment_commitment" />
          <bdi_clause
                entity_id="buyer"
                type="goal"
                id="PerformPayment_goal" />
          <bdi_clause
                entity_id="buyer"
                type="action"
                id="PerformPayment_action" />
     </message_content>
</team_message>
```


The buyer agent attempts the perform payment action relative to the goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_25">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="buyer">
          <recipient_list>
                <recipient agent_id="merchant" />
                <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
                truth_value="true"
                bdi_clause_id="PerformPayment_attempt"
                type="added"
                id="PerformPayment" />
          <bdi_clause
                action_id="PerformPayment_action"
                entity_id="buyer"
                commitment_id="PerformPayment_commitment"
                type="intention"
                id="PerformPayment_intention" />
          <bdi_clause
                entity_id="buyer"
                goal_id="PerformPayment_goal"
                type="commitment"
                id="PerformPayment_commitment" />
          <bdi_clause
                entity_id="buyer"
                type="goal"
                id="PerformPayment_goal" />
          <bdi_clause
```

```
                entity_id="buyer"
                type="attempt"
                intention_id="PerformPayment_intention"
                id="PerformPayment_attempt" />
          <bdi_clause
                entity_id="buyer"
                type="action"
                id="PerformPayment_action" />
     &lt;/message_content&gt;
&lt;/team_message&gt;
```

The buyer agent announces that the action of performing payment has been executed successfully

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
<team_message
     team_id="trade_team"
     message_id="MESSAGE_26">
     <meta_info
          content_type="framework-implementation" />
     &lt;delivery_info sender_id="buyer"&gt;
          &lt;recipient_list&gt;
               &lt;recipient agent_id="merchant" /&gt;
               &lt;recipient agent_id="delivery" /&gt;
          &lt;/recipient_list&gt;
          &lt;reply_to_list&gt;&lt;/reply_to_list&gt;
     &lt;/delivery_info&gt;
     &lt;message_content&gt;
          <message_clause
               truth_value="true"
               bdi_clause_id="PerformPayment"
               type="achieved" id="PerformPayment" />
          <bdi_clause
               entity_id="buyer"
               type="action"
               id="PerformPayment" />
     &lt;/message_content&gt;
&lt;/team_message&gt;
```

The merchant agent intends the receive payment action as part of the joint intention to the goal of performing trade

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
<team_message
     team_id="trade_team"
     message_id="MESSAGE_27">
     <meta_info
          content_type="framework-implementation" />
```

```
      <delivery_info sender_id="merchant">
            <recipient_list>
                  <recipient agent_id="buyer" />
                  <recipient agent_id="delivery" />
            </recipient_list>
            <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
            <message_clause
                  truth_value="true"
                  bdi_clause_id="ReceivePayment_intention"
                  type="added"
                  id="ReceivePayment" />
            <bdi_clause
                  entity_id="merchant"
                  type="action"
                  id="ReceivePayment_action" />
            <bdi_clause
                  action_id="ReceivePayment_action"
                  entity_id="merchant"
                  commitment_id="ReceivePayment_commitment"
                  type="intention"
                  id="ReceivePayment_intention" />
            <bdi_clause
                  entity_id="merchant"
                  type="goal"
                  id="ReceivePayment_goal" />
            <bdi_clause
                  entity_id="merchant"
                  goal_id="ReceivePayment_goal"
                  type="commitment"
                  id="ReceivePayment_commitment" />
      </message_content>
</team_message>
```

The merchant agent attempts the receive payment action relative to the goal of performing trade

```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_28">
      <meta_info
            content_type="framework-implementation" />
      <delivery_info sender_id="merchant">
            <recipient_list>
                  <recipient agent_id="buyer" />
                  <recipient agent_id="delivery" />
            </recipient_list>
            <reply_to_list></reply_to_list>
      </delivery_info>
```

```
     <message_content>
          <message_clause
               truth_value="true"
               bdi_clause_id="ReceivePayment_attempt"
               type="added"
               id="ReceivePayment" />
          <bdi_clause
               entity_id="merchant"
               type="attempt"
               intention_id="ReceivePayment_intention"
               id="ReceivePayment_attempt" />
          <bdi_clause
               entity_id="merchant"
               type="action" id="ReceivePayment_action"
/>
          <bdi_clause
               action_id="ReceivePayment_action"
               entity_id="merchant"
               commitment_id="ReceivePayment_commitment"
               type="intention"
               id="ReceivePayment_intention" />
          <bdi_clause
               entity_id="merchant"
               type="goal"
               id="ReceivePayment_goal" />
          <bdi_clause
               entity_id="merchant"
               goal_id="ReceivePayment_goal"
               type="commitment"
               id="ReceivePayment_commitment" />
     </message_content>
</team_message>
```


The merchant agent announces that the action of receiving payment has been executed successfully


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_29">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id=" merchant ">
          <recipient_list>
               <recipient agent_id="buyer" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
```

```
                truth_value="true"
                bdi_clause_id="ReceivePayment"
                type="achieved"
                id="ReceivePayment" />
          <bdi_clause
                entity_id="merchant"
                type="action"
                id=" ReceivePayment " />
     </message_content>
</team_message>
```


The delivery agent intends the deliver merchandise action as part of the joint intention to the goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_30">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="delivery">
          <recipient_list>
               <recipient agent_id="buyer" />
               <recipient agent_id="merchant" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"

     bdi_clause_id="DeliverMerchandise_intention"
               type="added"
               id="DeliverMerchandise" />
          <bdi_clause
               action_id="DeliverMerchandise_action"
               entity_id="delivery"
               commitment_id =
                         "DeliverMerchandise_commitment"
               type="intention"
               id="DeliverMerchandise_intention" />
          <bdi_clause
               entity_id="delivery"
               goal_id="DeliverMerchandise_goal"
               type="commitment"
               id="DeliverMerchandise_commitment" />
          <bdi_clause
               entity_id="delivery"
               type="goal"
               id="DeliverMerchandise_goal" />
          <bdi_clause
```

```
                entity_id="delivery"
                type="action"
                id="DeliverMerchandise_action" />
     </message_content>
</team_message>
```


The delivery agent attempts the deliver merchandise action relative to the goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_31">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="delivery">
          <recipient_list>
               <recipient agent_id="buyer" />
               <recipient agent_id="merchant" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
               bdi_clause_id="DeliverMerchandise_attempt"
               type="added"
               id="DeliverMerchandise" />
          <bdi_clause
               action_id="DeliverMerchandise_action"
                    entity_id="delivery"
               commitment_id =
                         "DeliverMerchandise_commitment"
               type="intention"
                    id="DeliverMerchandise_intention" />
          <bdi_clause
               entity_id="delivery"
               goal_id="DeliverMerchandise_goal"
               type="commitment"
               id="DeliverMerchandise_commitment" />
          <bdi_clause
               entity_id="delivery"
               type="goal"
               id="DeliverMerchandise_goal" />
          <bdi_clause
               entity_id="delivery"
               type="attempt"

     intention_id="DeliverMerchandise_intention"
```

```
                id="DeliverMerchandise_attempt" />
           <bdi_clause
                entity_id="delivery"
                type="action"
                id="DeliverMerchandise_action" />
     </message_content>
</team_message>
```

The delivery agent announces that the action of delivering merchandise has been executed successfully


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message team_id="trade_team"
message_id="MESSAGE_32">
     <meta_info content_type="framework-implementation"
/>
     <delivery_info sender_id="delivery">
           <recipient_list>
                <recipient agent_id="buyer" />
                <recipient agent_id="merchant" />
           </recipient_list>
           <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
           <message_clause
                truth_value="true"
                bdi_clause_id="DeliverMerchandise"
                type="achieved"
                id="DeliverMerchandise" />
           <bdi_clause
                entity_id="delivery"
                type="action"
                id="DeliverMerchandise" />
     </message_content>
</team_message>
```


The buyer agent intends the receive merchandise action as part of the joint intention to the goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_33">
     <meta_info
           content_type="framework-implementation" />
     <delivery_info sender_id="buyer">
           <recipient_list>
                <recipient agent_id="merchant" />
```

```
            <recipient agent_id="delivery" />
        </recipient_list>
        <reply_to_list></reply_to_list>
    </delivery_info>
    <message_content>
        <message_clause
            truth_value="true"

    bdi_clause_id="ReceiveMerchandise_intention"
            type="added"
            id="ReceiveMerchandise" />
        <bdi_clause
            entity_id="buyer"
            goal_id="ReceiveMerchandise_goal"
            type="commitment"
            id="ReceiveMerchandise_commitment" />
        <bdi_clause
            action_id="ReceiveMerchandise_action"
            entity_id="buyer"
            commitment_id =
                    "ReceiveMerchandise_commitment"
            type="intention"
            id="ReceiveMerchandise_intention" />
        <bdi_clause
            entity_id="buyer"
            type="action"
            id="ReceiveMerchandise_action" />
        <bdi_clause
            entity_id="buyer"
            type="goal"
            id="ReceiveMerchandise_goal" />
    </message_content>
</team_message>
```

The buyer agent attempts the receive merchandise action relative to the goal of performing trade

```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
    team_id="trade_team"
    message_id="MESSAGE_34">
    <meta_info
        content_type="framework-implementation" />
    <delivery_info sender_id="buyer">
        <recipient_list>
            <recipient agent_id="merchant" />
            <recipient agent_id="delivery" />
        </recipient_list>
        <reply_to_list></reply_to_list>
    </delivery_info>
    <message_content>
```

```
          <message_clause
               truth_value="true"
               bdi_clause_id="ReceiveMerchandise_attempt"
               type="added"
               id="ReceiveMerchandise" />
          <bdi_clause
               entity_id="buyer"
               goal_id="ReceiveMerchandise_goal"
               type="commitment"
               id="ReceiveMerchandise_commitment" />
          <bdi_clause
               action_id="ReceiveMerchandise_action"
               entity_id="buyer"
               commitment_id =
                         "ReceiveMerchandise_commitment"
               type="intention"
               id="ReceiveMerchandise_intention" />
          <bdi_clause
               entity_id="buyer"
               type="attempt"

     intention_id="ReceiveMerchandise_intention"
               id="ReceiveMerchandise_attempt" />
          <bdi_clause
               entity_id="buyer"
               type="action"
id="ReceiveMerchandise_action" />
          <bdi_clause
               entity_id="buyer"
               type="goal"
               id="ReceiveMerchandise_goal" />
     </message_content>
</team_message>
```


The buyer agent announces that the action of receiving merchandise has been executed successfully


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_35">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="buyer">
          <recipient_list>
               <recipient agent_id="merchant" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
```

```
            <message_clause
                  truth_value="true"
                  bdi_clause_id="ReceiveMerchandise"
                  type="achieved"
                  id="ReceiveMerchandise" />
            <bdi_clause
                  entity_id="buyer"
                  type="action"
                  id="ReceiveMerchandise" />
      </message_content>
</team_message>
```

The merchant agent intends the settlement of merchant's part of the trade as part of the joint intention to the goal of performing trade

```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_36">
      <meta_info
            content_type="framework-implementation" />
      <delivery_info sender_id="merchant">
            <recipient_list>
                  <recipient agent_id="buyer" />
                  <recipient agent_id="delivery" />
            </recipient_list>
            <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
            <message_clause
                  truth_value="true"
                  bdi_clause_id =
                            "MerchantCommitTrans_intention"
                  type="added"
                  id="MerchantCommitTrans" />
            <bdi_clause
                  entity_id="merchant"
                  type="action"
                  id="MerchantCommitTrans_action" />
            <bdi_clause
                  entity_id="merchant"
                  type="goal"
                  id="MerchantCommitTrans_goal" />
            <bdi_clause
                  action_id="MerchantCommitTrans_action"
                  entity_id="merchant"
                  commitment_id =
                            "MerchantCommitTrans_commitment"
                  type="intention"
```

```
               id="MerchantCommitTrans_intention" />
          <bdi_clause
               entity_id="merchant"
               goal_id="MerchantCommitTrans_goal"
               type="commitment"
               id="MerchantCommitTrans_commitment" />
     </message_content>
</team_message>
```

The merchant agent attempts the settlement of merchant's part of the trade relative to the goal of performing trade

```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_37">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="merchant">
          <recipient_list>
               <recipient agent_id="buyer" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"

     bdi_clause_id="MerchantCommitTrans_attempt"
               type="added"
               id="MerchantCommitTrans" />
          <bdi_clause
               entity_id="merchant"
               type="action"
               id="MerchantCommitTrans_action" />
          <bdi_clause
               entity_id="merchant"
               type="goal"
               id="MerchantCommitTrans_goal" />
          <bdi_clause
               action_id="MerchantCommitTrans_action"
               entity_id="merchant"
               commitment_id =
                         "MerchantCommitTrans_commitment"
               type="intention"
               id="MerchantCommitTrans_intention" />
          <bdi_clause
               entity_id="merchant"
               goal_id="MerchantCommitTrans_goal"
               type="commitment"
```

```
                id="MerchantCommitTrans_commitment" />
          <bdi_clause
                entity_id="merchant"
                type="attempt"

     intention_id="MerchantCommitTrans_intention"
                id="MerchantCommitTrans_attempt" />
     </message_content>
</team_message>
```


The merchant agent announces that the settlement of merchant's part of the trade has been executed successfully


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_38">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="merchant">
          <recipient_list>
                <recipient agent_id="buyer" />
                <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
                truth_value="true"
                bdi_clause_id="MerchantCommitTrans"
                type="achieved"
                id="MerchantCommitTrans" />
          <bdi_clause
                entity_id="merchant"
                type="action"
                id="MerchantCommitTrans" />
     </message_content>
</team_message>
```


The buyer agent intends the settlement of buyer's part of the trade as part of the joint intention to the goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team"
     message_id="MESSAGE_39">
     <meta_info
          content_type="framework-implementation" />
```

```
     <delivery_info sender_id="buyer">
          <recipient_list>
               <recipient agent_id="merchant" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
     <message_content>
          <message_clause
               truth_value="true"
               bdi_clause_id="BuyerCommitTrans_intention"
               type="added"
               id="BuyerCommitTrans" />
          <bdi_clause
               entity_id="buyer"
               goal_id="BuyerCommitTrans_goal"
               type="commitment"
               id="BuyerCommitTrans_commitment" />
          <bdi_clause
               entity_id="buyer"
               type="action"
               id="BuyerCommitTrans_action" />
          <bdi_clause
               entity_id="buyer"
               type="goal"
               id="BuyerCommitTrans_goal" />
          <bdi_clause
               action_id="BuyerCommitTrans_action"
               entity_id="buyer"

     commitment_id="BuyerCommitTrans_commitment"
               type="intention"
               id="BuyerCommitTrans_intention" />
     </message_content>
</team_message>
```


The buyer agent attempts the settlement of buyer’s part of the trade relative to the goal of performing trade


```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
     team_id="trade_team" message_id="MESSAGE_40">
     <meta_info
          content_type="framework-implementation" />
     <delivery_info sender_id="buyer">
          <recipient_list>
               <recipient agent_id="merchant" />
               <recipient agent_id="delivery" />
          </recipient_list>
          <reply_to_list></reply_to_list>
     </delivery_info>
```

```
      <message_content>
            <message_clause
                  truth_value="true"
                  bdi_clause_id="BuyerCommitTrans_attempt"
                  type="added"
                  id="BuyerCommitTrans" />
            <bdi_clause
                  entity_id="buyer"
                  goal_id="BuyerCommitTrans_goal"
                  type="commitment"
                  id="BuyerCommitTrans_commitment" />
            <bdi_clause
                  entity_id="buyer"
                  type="attempt"
                  intention_id="BuyerCommitTrans_intention"
                  id="BuyerCommitTrans_attempt" />
            <bdi_clause
                  entity_id="buyer"
                  type="action"
                  id="BuyerCommitTrans_action" />
            <bdi_clause
                  entity_id="buyer"
                  type="goal"
                  id="BuyerCommitTrans_goal" />
            <bdi_clause
                  action_id="BuyerCommitTrans_action"
                  entity_id="buyer"

      commitment_id="BuyerCommitTrans_commitment"
                  type="intention"
                  id="BuyerCommitTrans_intention" />
      </message_content>
</team_message>
```

The buyer agent announces that the settlement of merchant's part of the trade has been executed successfully

```
<?xml version="1.0" encoding="UTF-8"?>
<team_message
      team_id="trade_team"
      message_id="MESSAGE_41">
      <meta_info
            content_type="framework-implementation" />
      <delivery_info sender_id="buyer">
            <recipient_list>
                  <recipient agent_id="merchant" />
                  <recipient agent_id="delivery" />
            </recipient_list>
            <reply_to_list></reply_to_list>
      </delivery_info>
      <message_content>
```

```
        <message_clause
            truth_value="true"
            bdi_clause_id="BuyerCommitTrans"
            type="achieved"
            id="BuyerCommitTrans" />
        <bdi_clause
            entity_id="buyer"
            type="action"
            id="BuyerCommitTrans" />
    &lt;/message_content&gt;
&lt;/team_message&gt;
```

Trade agents individually reason that the trade action has succeeded and hence, the goal of performing trade has succeeded

merchant: *************** Perform Trade Succeeded ****************
delivery: *************** Perform Trade Succeeded ****************
merchant: *************** Perform Trade Succeeded ****************

Bibliography

[1] G. Weiss, Editor, "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence", MIT Press, 1999.

[2] J. Ferber, "Multi-agent Systems: An Introduction to Distributed Artificial Intelligence", Addison-Wesley, 1999.

[3] Tim Finnin, Yannis Labrou and James Mayfield, "KQML as an Agent Communication Language" in "Software Agents", editor Jeffery. Bradshaw, AAAI Press/MIT Press, 1997.

[4] M. Huhns and L. Stephens, "Multiagent Systems and Societies of Agents" in "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence", editor G. Weiss, MIT Press, 1999.

[5] M. Singh, A. Rao and M. Georgeff in "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence", editor G. Weiss, MIT Press, 1999.

[6] M. Nowostawski, M. Purvis and S. Cranefield, "Modeling and Visualizing Agent Conversations", Proceedings of the fifth international conference on Autonomous agents, July 2001.

[7] N. Jennings, "Commitments and Conventions: The Foundation of Coordination in Multi-Agent Systems", The Knowledge Engineering Review, 8 (3), 223-250, 1993.

[8] N. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra and M. Wooldridge, "Automated negotiation: prospects, methods and challenges" Int. J. of Group Decision and Negotiation 10 (2) 199-215, 2001.

[9] M. Dastani, J. Hulstijn and L. Torre, "Negotiation Protocols and Dialogue Games", Proceedings of the fifth international conference on Autonomous agents, July 2001.

[10] K. Carley and L. Gasser, "Computational Organization Theory" in "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence", editor G. Weiss, MIT Press, 1999.

[11] E. Durfee, “Distributed Problem Solving and Planning” in "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence", editor G. Weiss, MIT Press, 1999.

[12] M. Tambe, "Towards Flexible Teamwork", Journal of Artificial Intelligence Research, 1997.

[13] B. Wilsker, "A Study of Multi-Agent Collaboration Theories", ISI Research Report, 1996.

[14] P. Cohen and H. Levesque, "Teamwork", Nous 25(4), Special Issue on Cognitive Science and Artificial Intelligence, pp. 487-512, 1991.

[15] B. Dunin-Keplicz and R. Verbrugge, "Collective Intentions", Annals of Mathematics and Artificial Intelligence, March 2001.

[16] M. Wooldridge and N. Jennings, "Cooperative Problem Solving" Journal of Logic and Computation 9 (4) 563-592, 1999.

[17] N. Jennings, E. Mamdani, I. Laresgoiti, J. Perez and J. Corera, "GRATE: A General Framework for Cooperative Problem Solving", IEE-BCS Journal of Intelligent Systems Engineering, 1 (2), 102-114, 1992.

[18] M. Brinn, T. Carrico, N. Combs , "Every agent a web server, every agent community an intranet…", Proceedings of the fifth international conference on Autonomous agents, Montreal, Quebec, Canada, 14.Pages: 226 - 227, 2001.

[19] Peter Stone, Michael L. Littman, Satinder Singh and Michael Kearns, “An Adaptive Autonomous Bidding Agent”, Proceedings of the fifth international conference on Autonomous agents, July 2001.

[20] F. Zambonelli, N. Jennings, A. Omicini and M. Wooldridge, "Agent-Oriented Software Engineering for Internet Applications" in Coordination of Internet Agents (eds. A. Omicini, F. Zambonelli, M. Klusch and R. Tolksdorf) Springer Verlag, 326-346, 2001.

[21] M. Wooldridge and N. Jennings, "Intelligent Agents: Theory and Practice", The Knowledge Engineering Review, 10 (2), pp. 115-152, 1995.

[22] A. Lomuscio, M. Wooldridge and N. Jennings, "A Classification Scheme for Negotiation in Electronic Commerce" in Agent-Mediated Electronic Commerce:

A European AgentLink Perspective (eds. F. Dignum and C. Sierra), Springer Verlag, 19-33, 2001.

[23] Q. Chen, M. Hsu, U. Dayal and M. Griss, "Multi-Agent Cooperation, Dynamic Workflow and XML for E-Commerce Automation", Proceedings of the fourth international conference on Autonomous agents, June 2000.

[24] A. Glass and B. Grosz, "Socially Conscious Decision-Making", Proceedings of the fourth international conference on Autonomous agents', Barcelona, Spain, 2000.

[25] B. Grosz and S. Kraus, "Collaborative Plans for Complex Group, Action", Artificial Intelligence, 86(2), pp. 269-357, 1996.

[26] D. Sullivan, B. Grosz and S. Kraus, “Intention Reconciliation by Collaborative Agents”, Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000), IEEE Computer Society Press, 2000.

[27] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. Asada, "The RoboCup Synthetic Agent Challenge 97", International Joint Conference on Artificial Intelligence (IJCAI97), 1997.

[28] J. E. Doran, S. Franklin, N. Jennings and T. J. Norman, "On cooperation in multi-agent systems". The Knowledge Engineering Review, 12(3), 309-314, 1997.

[29] J. Graham and K. Decker, "Towards Distributed, Environment Centered Agent Framework", Appearing in Intelligent Agents IV, Agent Theories, Architectures, and Languages, Nicholas Jennings and Yves Lesperance, Editors, Springer-Verlag, 2000.

[30] J. Huang, N. Jennings, and J. Fox, "Cooperation in Distributed Medical Care", Proc. Int. Conf. on Cooperative Information Systems (CoopIS-94), Toronto, Canada, 1994, 255-263.

[31] L. M. Hogg and N. Jennings, "Socially Rational Agents", in Proc. AAAI Fall symposium on Socially Intelligent Agents, Boston, Mass., November 8-10, 61-63, 1997.

[32] M. Barbuceanu, T. Gray and S. Mankovski, "Coordinating with Obligations", Proceedings of the 2nd International Conference on Autonomous Agents (AGENTS-98), pp. 62-69, ACM Press, May 9-13 1998.

[33] M. Schut and M. Wooldridge, "Intention reconsideration in complex environments", Proceedings of the fourth international conference on Autonomous agents, Barcelona, Spain, 2000.

[34] M. Tambe, "Agent architectures for flexible, practical teamwork", National Conference on Artificial Intelligence (AAAI-97), 1997.

[35] M. Veloso, P. Stone, and M. Bowling, "Anticipation: A Key for Collaboration in a Team of Agents", SPIE Sensor Fusion and Decentralized Control in Robotic Systems II (SPIE'99), 1999.

[36] M. Wooldridge and N. Jennings, "Cooperative Problem Solving" Journal of Logic and Computation 9 (4) 563-592, 1999.

[37] M. Wooldridge and N. Jennings, "Towards a Theory of Cooperative Problem Solving", Proc. Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-94), Odense, Denmark, 15-26, 1994.

[38] M. Wooldridge, "Intelligent Agents" In G. Weiss, editor: Multiagent Systems, The MIT Press, April 1999.

[39] N. Jennings and M. Wooldridge, "Applications of Intelligent Agents" in Agent Technology: Foundations, Applications, and Markets (eds. N. Jennings and M. Wooldridge) 3-28, 1998.

[40] N. Jennings and M. Wooldridge, "Applying Agent Technology, Applied Artificial Intelligence", An International Journal, Taylor & Francis London, 9 (4) 1995, 351-361.

[41] N. Jennings and M. Wooldridge, "Software Agents", IEE Review, January, 17-20, 1996.

[42] N. Jennings, "A Knowledge Level Approach to Collaborative Problem Solving", Proc. AAAI Workshop on Cooperation Among Heterogeneous Intelligent Agents, San Jose, USA, 55-64, 1992.

[43] N. Jennings, "Agent Software", Proc. UNICOM Seminar on Agent Software, London, UK, 1995, 12-27.

[44] N. Jennings, "Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems using Joint Intentions", Artificial Intelligence, Vol. 75 (2), 1995, 195-240.

[45] N. Jennings, "Controlling Cooperative Problem Solving Using Joint Intentions", AI Communications 6 (3&4), 247-248, 1993.

[46] N. Jennings, "Coordination Techniques for Distributed Artificial Intelligence", in Foundations of Distributed Artificial Intelligence (eds. G. M. P. O'Hare and N. Jennings), Wiley, 187-210, 1996.

[47] N. Jennings, "Coordination Through Joint Intentions in Industrial Multi-Agent Systems", AI Magazine, 14 (4), 1993, 79-80.

[48] N. Jennings, "On Being Responsible", Decentralized A. I. 3, (eds. E. Werner & Y. Demazeau), North Holland Publishers, 93-102, 1992.

[49] N. Jennings, "Specification and Implementation of a Belief-Desire-Joint-Intention Architecture for Collaborative Problem Solving", Int. Journal of Intelligent and Cooperative Information Systems, 2 (3), 289-318, 1993.

[50] N. Jennings, "Towards a Cooperation Knowledge Level for Collaborative Problem Solving", Proc. 10th European Conf. on Artificial Intelligence (ECAI-92), Vienna, Austria, 1992, 224-228.

[51] N. Jennings, J. M. Corera, I. Laresgoiti, "Developing Industrial Multi-Agent Systems", (Invited Paper), First International Conference on Multi-Agent Systems (ICMAS'95), San Francisco, CA., June 12-14, 1995, 423-430.

[52] N. Jennings, K. Sycara and M. Wooldridge, "A Roadmap of Agent Research and Development", Int Journal of Autonomous Agents and Multi-Agent Systems 1 (1) 7-38, 1998.

[53] N. Jennings, S. Parsons, C. Sierra and P. Faratin, "Automated Negotiation", Proc. 5th Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Systems (PAAM-2000), Manchester, UK, 23-30, 2000.

[54] N. Karacapilidis, and P. Moraitis, "Intelligent agents for an artificial market system", Proceedings of the Fifth International Conference on Autonomous Agents, pp. 592-599, ACM Press, May 2001.

[55] P. Panzarasa and N. Jennings, "Negotiation and joint commitments in multi-agent systems", Sozionik aktuell 3 65-81, 2001. [Also appearing in: Proc. 2nd Int Workshop on Modelling Artificial Societies and Hybrid Organisations, Vienna, Austria]

[56] P. Panzarasa and N. Jennings, "Social influence and the generation of joint mental attitudes in multi-agent systems", Proc. 4th Int Eurosim Congress on Simulating Organisational Processes, Delft, The Netherlands, 2001.

[57] R. Hill, J. Chen, J. Gratch, P. Rosenbloom, and M. Tambe, "Intelligent agents for the synthetic battlefield: A company of rotary wing aircraft", Innovative Applications of Artificial Intelligence (IAAI-97), 1997.

[58] S. Kalenka and N. Jennings, "Socially Responsible Decision Making by Autonomous Agents" in Cognition, Agency and Rationality (eds K. Korta, E. Sosa, X. Arrazola) Kluwer 135-149, 1999.

ملخص الرسالة

يمثل الوكيل كيانا تصورياً أو ماديا يتمتع بالقدرة على إنتاج نسخ مشابهة له والاتصال مع وكلاء آخرين، كما يتمتع بمجموعة من المهارات ويمكنه عمل مجموعة من الخدمات، كما يمتلك موارد حاسوبية ويتحرك بناءً على وجود أهداف خاصة أو غرائز داخلية.

والأنظمة متعددة الوكلاء يمكن تعريفها كشبكة من الوكلاء يستطيع أفرادها العمل معاً لحل مشكلات تتجاوز حدود القدرات الفردية والمعرفية للوكيل. و يعرف التعاون في أنظمة متعددة الوكلاء كشكل من أشكال المشاركة يتميز بوجود تفاهمات أحادية وبنظرة مشتركة للمشكلة المطلوب حلها بين كل الوكلاء المشاركين. و يحدث التعاون بين فريق من أعضاء الوكلاء الذين يتميزون بالقدرة على استخدام البرهنة كأدوات لصنع القرار.

وتوفر النمذجة المبنية على مفهوم الوكالة إمكانية للتجريد أعلى من تلك التي توفرها النمذجة المعتمدة على مفهوم الشيئية، الأمر الذي يجعلها اكثر مناسبة لتطوير البرامج في الأنظمة المفتوحة والتي تتميز بخصائص التركيب وعدم التشابه في مكونات النظام، كما يتميز النظام المفتوح بعدم وجود إمكانية للجزم مسبقا بهيكليته أو مكوناته. ويؤدى دعم نموذج الوكالة للنظم المفتوحة إلى دعم التنوع في الأشكال المختلفة لهياكل نظم التشغيل والشبكيات والهياكل الداخلية لأنظمة الوكلاء وكذلك بروتوكولات التفاعل بين الوكلاء. ويؤثر هذا التنوع في أسلوب إنشاء إطار عام للتفاعل لدعم الأنظمة المفتوحة بسبب عدم إمكانية الجزم بوجود هيكل تنظيمي ثابت أو هيكلية داخلية لشكل معينة للوكيل أو لبروتوكولات التفاعل المستخدمة.

في هذا البحث نقدم تطويرا لإطار عمل مقترح لدعم التعاون في الأنظمة متعددة الوكلاء غير المتشابهين، و يبنى الإطار المقترح على نموذجين رياضيين عموميين للتعاون وحل المشكلات، كما يبنى على رؤية طبقية لهيكلية المحادثات بين الوكلاء كأحد أشكال التفاعل. ويتمثل إطار العمل المقترح في مجموعة من الوصفات التي يقوم مطور أنظمة الوكلاء باتباعها لضمان نجاح التعاون بين وكلاء النظام متعدد الوكلاء. ولا توجد أي اعتمادية لهذا الأسلوب المقترح على أي شكل تنظيمي معين للأنظمة متعددة الوكلاء، كما لا يعتمد الأسلوب المقترح على هيكلية داخلية معينة أو على أي من بروتوكولات التفاعل المستخدمة. ويقوم المنهج المستخدم على الفصل بين الإطار العام للتعاون وبين الأشكال الممكنة لتنفيذ هذا الإطار.

وأخيراً، نقوم في هذا البحث بتنفيذ أحد التطبيقات الممكنة للإطار العام والتي نستخدمها لاحقا في هذا البحث في دراسة حالة لنظام متعدد الوكلاء لدعم تطبيقات التجارة الإلكترونية.

إطار لتعاون الوكلاء في الأنظمة متعددة الوكلاء

إعداد

محمد جمال الدين عطواني

إشراف

د. عبد العزيز خميس	**د. مجدي أبو العلا**
قسم علوم الحاسب و المعلومات	قسم علوم الحاسب و المعلومات
جامعة القاهرة	أكاديمية السادات للعلوم الإدارية

قدمت هذه الرسالة إستكمالا متطلبات الماجستير في علوم الحاسب قسم علوم الحاسب و المعلومات, معهد الدراسات و البحوث الإحصائية, جامعة القاهرة

مايو 2002