Utrecht University

Institute of Information and Computing Sciences

Identifying Roles in Multi-Agent Systems

by overhearing

Maurice Vanderfeesten

May, 2006

INF-SCR-05-62

Supervisors:

Dr. Virginia Dignum,
Dr. Ir. Robbert-Jan Beun,
Drs. Jurriaan van Diggelen.

Author

Maurice Vanderfeesten

Utrecht University
Institute of Information and Computing Sciences
Department: Content and Knowledge Engineering

Study: Information Science
Student number: 9908927
E-mail: mpjpvand@cs.uu.nl /
maurice.vanderfeesten@gmail.com

Universiteit Utrecht

Utrecht University

Main thesis supervisor Dr. V. Dignum

Institute of Information and Computing Sciences
Department: Content and Knowledge Engineering
E-mail: virginia@cs.uu.nl

Second thesis supervisor Dr. Ir. R. Beun

Institute of Information and Computing Sciences
Department: Content and Knowledge Engineering
E-mail: rj@cs.uu.nl

Third thesis supervisor Drs. J. van Diggelen

Institute of Information and Computing Sciences
Department: Machine Learning
E-mail: jurriaan@cs.uu.nl

Space for comments

Abstract

In our daily life we are confronted with people in our society who act in different roles. To communicate and to understand each other it is important to recognize these roles. If you understand the implications of a role, you can better understand what this person tries to tell to you.

It is not always clear what role a person enacts. Often we match certain properties of persons to roles we already know by resembling common verbal and nonverbal expressions. In this thesis we focus on verbal communicative expressions, in particular the communicative acts supported as a standard by the Foundation for Intelligent Physical Agents (FIPA). These expressions follow social rules. These rules are collected in a protocol. Therefore we assume that a role is defined by typical expressions.

In a society of software agents we are testing two methods to recognize roles. These methods we define as deductive and inductive reasoning processes. In the deductive method a dialog between two software agents is validated according to a Document Type Definition (DTD), where dialogs are formatted in an extendible mark-up language (XML) document. In the inductive method a Naive Bayes Classifier determines the role from a dialog between two software agents.

In this thesis it is assumed that two different organizations contain in essence similar roles, but the way these roles express themselves can be different. These variations we call “dialects”. We assume that the essence of a similar role in different societies remains the same, which means in this thesis that every similar role tries to achieve a similar objective.

The objective a role-enacting agent has to achieve is an internal property of a role and in this thesis it is assumed that internal properties cannot be observed. The expressions the role-enacting agents use to communicate with each other, however, can be observed. These sets of communicative expressions appear in a dialog.

In this thesis we test the deductive and inductive reasoning process to provide a recommendation in order to recognize roles in a multi-agent system by overhearing. Both methods used contain a learning and a testing phase. In the learning phase an observer learns in one organization to recognize roles by overhearing conversations between two role-enacting agents. In the testing phase this observer applies this learned material to recognize the roles of two role-enacting agents in another organization.

To test both methods, we started with one original dialog as a control dialog and created five other dialogs, dialects, by varying the verbal communicative expressions. To pass the test means a role has been recognized from the observed dialog. To fail the test no role has been

classified. The results are that the deductive method is too inflexible; only the control dialog passed the test. The dialects for this method were not recognized at all. The inductive method considers all six dialogs passed the test.

At this point the inductive method is recommended for recognizing roles. Future research has to be made to calculate the quality of the outcome of the classification.

Using a role recognition method in multi-agent systems can be used as a tool to expand social relations (semi-)autonomously beyond the borders of an organization. This can give an agent the ability to increase their working domain where they can have dialogs with role-enacting agents from other organizations where prior relationships did not exist.

Acknowledgements

"*Something that has always puzzled me all my life is why, when I am in special need of help, the good deed is usually done by somebody on whom I have no claim.*"
-William Feather

Virgina was my main supervisor. She motivated me with her enthusiasm about the contribution subject of the thesis would deliver to the scientific agent community. Her lectures about knowledge management were my initial drive to make a contribution in this field.

Robber-Jan was my second supervisor. He provided very critical remarks and supported me to keep the thesis on a scientific track. Also he provided me with information about roles and using communication.

Jurriaan van Diggelen was my third supervisor. He entered in the final phase of the thesis process. He was very enthusiastically helping with tackling some writing problems. Also he helped me with the part about machine learning. Beside this he supported me in setting up the fundament of the thesis with an explorative character.

Special thanks go to Antal Reijnders and Scato Eggen who supported me writing this thesis in English, provided critical remarks and feedback. Also I would like to thank my friends, housemates, roommates from the Agent-lab, girl friend, and parents for their support during my study and in this final phase of writing this thesis.

Many thanks to all these people who motivated me finishing my thesis, due to their support and love.

Preface

Information Science started in the year 2000 as a new study at the Utrecht University in the Netherlands, and is part of the faculty Mathematics and Computer Science. This thesis is a graduate assignment and forms the termination of more than four years of study. It resulted in much acquired knowledge, although it is in fact a base for the future.

The idea for this research came from my involvement in theatre and at the same time attending to lectures about knowledge management. At the lectures I learned about distributed software agents who communicate with each other to share knowledge. When involved in theatre I was wondering about the fact how the theatre audience was able to understand what roles the actors play. And whether this understanding of roles could help distributed software agents to find each other to share their knowledge even better without human involvement.

List of terms and abbreviations

Blackboard	A webpage where all agents use to communicate. This is one central page where dialogs appear like on a chat channel.
Dialect	A variation on an official language.
DTD	Document Type Definition. This defines the structure an XML document should follow. This is used to validate the structure of XML documents.
FIPA	Foundation of Intelligent Physical Agents. This foundation has created a standard that describes an architecture for multi-agent systems how software agents should be organized and communicate.
HTML	Hyper Text Mark-up Language, A tag-based language to layout web pages.
KM	"Knowledge Management (KM) refers to the process of creating, codifying and disseminating knowledge within complex organizations, such as large companies, universities, and organizations for social and civil services." [Bonifacio, 2002]
Multi-agent system	A computer system or a network of computer systems where agents can communicate with each other. This communication is based on certain architecture. In this thesis we mainly use the OperA model. This is a theoretic model of an organization that describes how agents communicate based on the FIPA architecture. The implementation can be in for example the JAVA programming language.
Naïve Bayesian	A classification method based on the Bayes theorem. In Naive Bayesian classification variables are considered to be

classification	independent, not interdependent. More about this in section 2.5.2.
Role-enacting agent	A software agent who performs a role in a multi-agent system
Software agent	A software agent is a piece of software that simulates specific tasks a real person could do. Also referred as intelligent agent. This agent can make decisions autonomously. In a setting where more then one agent is operating cooperatively or competitively we speak of a multi-agent system. In a multi-agent system where agents act in different roles this is referred to as a multi-agent society or multi-agent organization.
Stereotyping	Stereotypes are considered to be a group concept, held by one social group about another. Stereotyping is based on: Simplification, Exaggeration or distortion, Generalization, Presentation of cultural attributes as being 'natural'. In computing, a stereotype is a concept in the Unified Modelling Language (UML), where it is used to encapsulate method behaviour such as <<constructor>> and <<getter>>. [wikipedia.org] In the thesis also the term social profiling is used.
XML	eXtendible Mark-up Language. A tag-based language in which content can be structured. This language is a standard for defining formal mark-up languages which can be readable for machines and humans. The naming and structure of the tags can be used freely or can be used as an international standard which requires the consensus of a community to create consensus.

List of Figures

List of Tables

Table of Content

1 Introduction

Saturday at noon you walk around the busy canals of Utrecht in the shopping area, as you walk by, you pick up a part of a conversation:

...
A: I have a fourteen year old son.
B: Well that's all right.
A: I also have a dog.
B: Oh, I'm sorry.
...
[Levinson, 1983]

The above conversation seems bizarre in isolation, but when embedded in an actual situation -A is trying to hire a room from landlord B- it seems natural and quite understandable. This demonstrates that role *recognition* may substantially improve the understanding of conversational contributions. The role appears to provide valuable *contextual information* to determine the meaning of the utterances.

Let's take another example. You walk into the classroom; it is the beginning of the semester and you have some questions for the teacher about the schedule. You don't know what he or she looks like. How do you know who the teacher is? Are you going to look for external appearance, such as grey hair? Are you going to look for certain behavioral aspects, such as a person who stands up and commands the other people to open their books at page one? Or are you going to disturb every single person by asking each of them if he or she is the teacher? We use our senses to observe expressions from other people and resemble these to expectations we have of the behavior of a role. In fact we match the observed behavior to the behavior one expects.

Recognizing the role someone enacts is usually a matter of stereotyping that is vital in daily life to get around and to get things done to meet your objective. In both examples to recognize stereotype roles we look for features that enable us to recognize an individual. An individual who enacts a role sends out *characteristic information* about the role he or she occupies. This can be accomplished by looking at the various verbal and nonverbal channels that provide this characteristic information. For example verbally on the auditive channel when you hear someone say: "Sir, you are under arrest" - the only role in society who can legally say such a thing is a police officer, or nonverbally on the visual channel that provides information that a person is likely to be a police officer if he or she is wearing a police uniform, hand coughs, a pistol and a badge.

1.1 Software agents

Clearly, being aware of the roles in society and organizations makes understanding each other easy. We know where to get information, we know who can do a particular job (like arresting people) and, as the first example shows us, it enables us to understand the meaning of a conversation.

In the electronic world individuals are represented by software agents and, to some extent, these agents show the same behavior as the individuals in our physical world. They show autonomous behavior, can communicate and since most of them are specialized, they play roles in a virtual sense.

1.2 Problem description

Like individuals in the physical world, electronic agents must be able to recognize roles. Software agents that cannot recognize a role will be constantly talking to the wrong agent, and will not be able to proactively engage a conversation to pursue its objective. (See Illustration 1)

To stress the importance of role recognition we use the field of knowledge management to provide an example.

"Knowledge Management (KM) refers to the process of creating, codifying and disseminating knowledge within complex organizations, such as large companies, universities, and organizations for social and civil services." [Bonifacio, 2002] The tradition in KM projects is to create large, homogeneous knowledge repositories. In this repository the knowledge is represented according to a single conceptual schema[1]. Using a single schema for the whole organization is incompatible with the nature of knowledge. This incompatibility lies in the mismatch between social form and technological architecture. "The social form is that each community uses its own conceptual schema to describe the world of phenomena in that specific domain. (See Illustration 1)" [Bonifacio, 2002]

[1] Conceptual schemas: schema's that represent shared conceptualisations of corporate knowledge, and enables communication and knowledge sharing across the entire organisation.

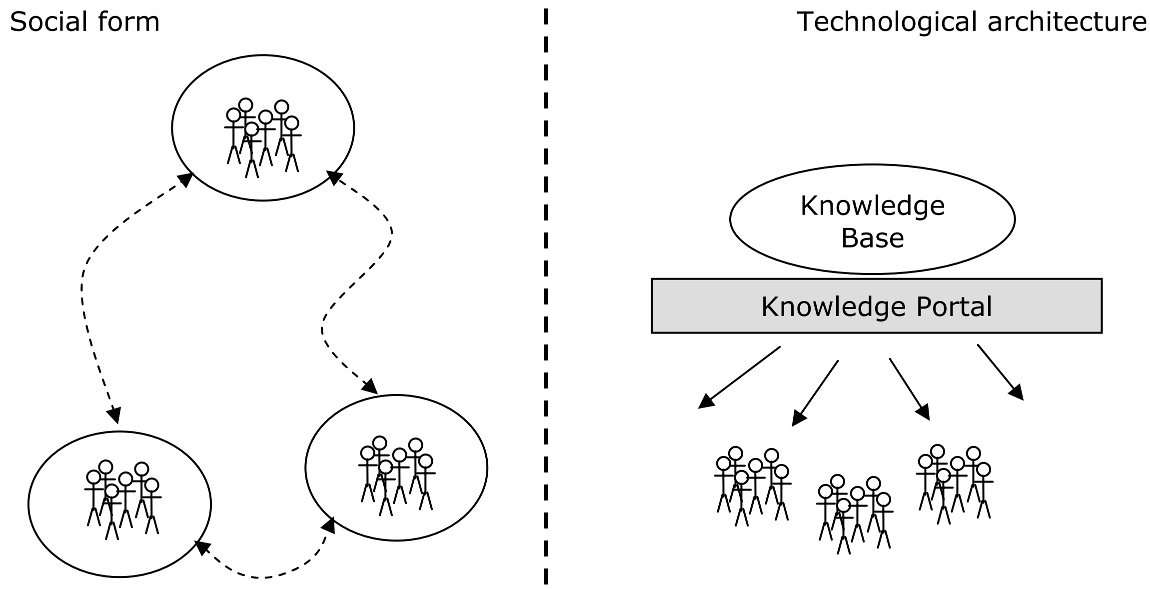


Illustration 1: Traditional knowledge management is lead through technology but creates a mismatch between the social form people normally communicate and share knowledge. [Bonifacio, 2002]

Bonifacio et al. describes an architecture that is more compatible with the social form. "The first idea is that knowledge should be autonomously managed where it is created and used, namely within each community. Autonomy means, for example, that each community should be allowed to build its own local conceptual schema.

The second idea is that each community has to create a link from its own conceptual schema to the schema of another community. These mappings can be defined by hand, but a KM system should support the creation of automatic (or semiautomatic) mappings. This requires the ability to compare, in a semantically relevant way, autonomously generated contexts.

The third idea points out that context matching is not enough. For human beings, knowledge sharing is often the result of a social process, in which many different cooperative strategies are used. The third idea underlying the proposed architecture is that the system should aim at to reproducing this social process." [Bonifacio, 2002] See also: [Davies, 1998]; [Elst, 2003]; [Bonifacio, 2003]; [Mathieu, 2002]

In this thesis our aim contributes to the second and third idea; comparing autonomously generated contexts by matching social processes (conversations) by finding agents that are socially compatible to one another. In a social community a software agent enacts a social role. To find agents with a matching social process, we look at similarities in conversation strategies of role-enacting agents from other domains.

Flexibility and scalability

Let's assume that social interaction between software agents appears distributed on the World Wide Web. Let's also assume that on the World Wide Web software agents can autonomously find social compatible parties. Then these properties, distribution and recognition, creates the flexibility and scalability to share knowledge that lies beyond the specific domain of a software agent in a corporate multi-agent system. [Davies, 1998]; [Elst, 2003]; [Bonifacio, 2003]; [Mathieu, 2002]

Providing an autonomous coordination mechanism for each agent, organization boundaries could disappear, and a group of agents can create a self-organizing virtual community of agents with expert abilities. In this virtual community, specialized agents can pursue either their own or other agents' objectives. The advantage will be that the objective of an agent can much faster be achieved when connected to this virtual community, than restricted to the boundaries of an organization. Being inside this community the pool of agents is much bigger then limited to the agents inside an organization. For example we can see this community as a web-forum where people come together to help each other.

The problem is when an agent in the community is called "BR2992" there is no way to tell that this agent is a real estate broker. Agents who perform a role are recognized for what they do or say, not for what their names are. Therefore we have to create a universal mechanism that can determine what role an agent performs. As stated before we will look at the compatibility of conversation strategies of a role-enacting agent.

1.3 Research question

This thesis has an explorative character; the following question will be investigated:

"How can a software agent recognize roles in a multi-agent system?"

The thesis statement is: Software agents are able to recognize the roles that other agents enact by overhearing the conversations between the other agents.

To answer this research question we setup several research tasks involving role recognition as social profiling[2]:

1. Investigate the definition of a role.

[2] Social profiling: Making a profile of social behaviour.

2. Investigate the definition of a multi-agent system that uses role-enacting agents.

3. Investigate the definition of communication as an expression for recognizing roles.

4. Investigate recognition methods.

5. Investigate the implementation of the recognition methods.

6. Create a method to compare the recognition methods.

    a. Create a specification that formally describes each method

    b. Create an implementation for testing each method.

Solutions for these research tasks will be appear throughout the following sections of the thesis.

1.4 Thesis Scenario

To create a mechanism that can determine the role an agent performs, we created an example we called the "thesis scenario". This scenario describes the scope of the problem that has been investigated. This is used to test our recognition methods described in section 3.

The thesis scenario is about an agent who has a question, but cannot get the answer inside his own organization. Therefore this agent is going to look for another agent in other organizations that can answer his question. In the other organizations he can only see what agents are saying to each other (like on a web-forum).

Illustration 2 shows an agent observing the dialogs of other agents, which we will call the observer agent. The agent that makes the observation is going to look for the side of dialogs the observing agent is interested in. Our assumption is that in a dialog each agent shows typical behavior in language. This typical behavior exposes the role that an agent performs. Also we assume that placing a role to an agent will help to create better understanding of a dialog.

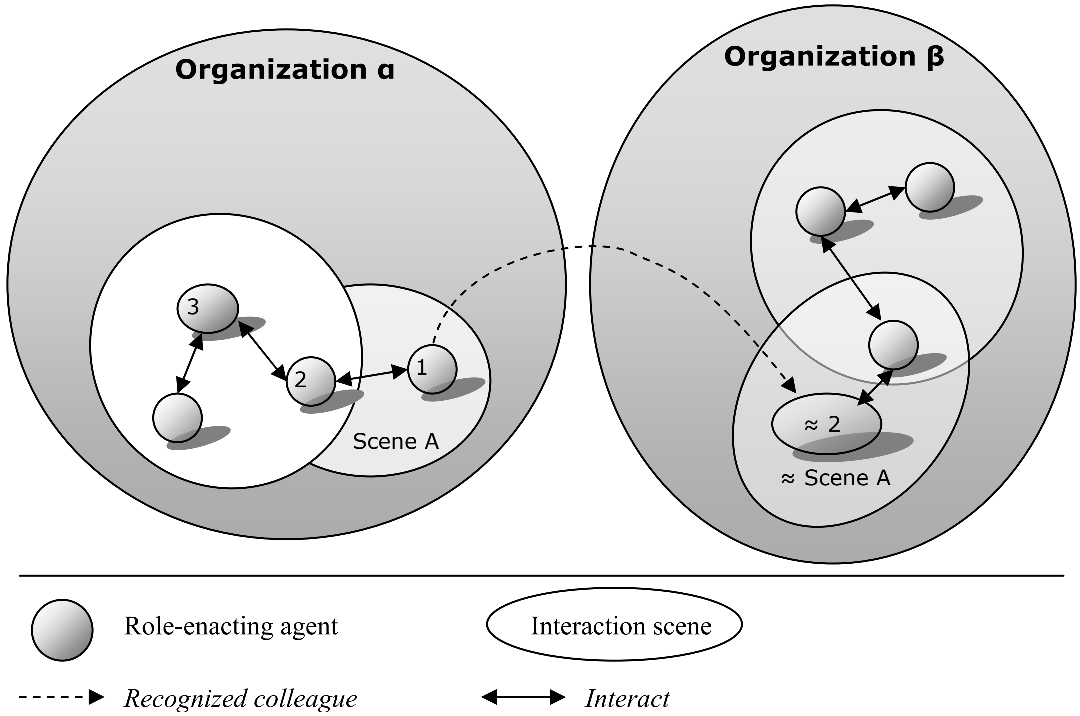


Illustration 2: Colleagues, that are role-enacting agents with similarities from different organizations, are able to meet their objective.

Illustration 2 shows how the *multi-agent system* for this scenario looks like. In this case the architecture of this system is based upon *the OperA model* from [Dignum, 2004]. We see two organizations. In these organizations communication between two or more role-enacting agents takes place within *interactions scenes*. A role-enacting agent is an agent who is currently occupying a role in an organization. An interaction scene is a place where two or more agents interact with each other to work on a goal set by the creators of the organization.

Role-enacting agent 1 from organization α regularly communicates with role-enacting agent 2 in interaction scene A. Agent 1 is not able to meet the objective that is set for his role. He is increasing his scope by looking for a colleague in organization β who resembles role-enacting agent 2.

To find a role, the observer agent 1 first learns to recognize role-enacting agent 2 from the conversations in scene A. Then the observer agent looks at the conversations between the role-enacting agents in organization β. From these observed conversations, agent 1 tries to find out which role-enacting agents show similarities in the conversations he has with role-enacting agent 2.

In the perspective of role recognition we find similarities with a drama play; most of the time when the play starts one does not know the objectives, or the name of the actor playing a role.

In the duration of the play one will be able to learn what role an actor enacts from the expressions that one observes: words, gesture, clothing, etc. [Elam, 1980]

To deliver more focus to the scenario some assumptions are made.

Assumptions

Interaction scenes define what roles are used, also what *protocols* role-enacting agents might use. A protocol is a set of rules that tells the agents how to communicate. Our assumption is that the given names to roles and protocols can differ in every organization and therefore looking for similarities in names will not be feasible to recognize agents the observer is interested in.

A role defines the objectives an agent has to meet and also tells the agent how to behave by given *norms*. [Dignum, 2004] Also we assume that objectives and norms are an internal matter of a role-enacting agent and are not visible to the outside world and therefore also not feasible for recognizing colleagues.

However, the expressions in verbal communication are visible on a blackboard[3]. These externalized expressions are visible for the observer agent. This means that it is a feasible utility to find similarities.

Another assumption is: a software agent is able to enact a role inside an organization. Role-enacting agents inside this organization communicate within interaction scenes[4]. Role-enacting agents use protocols[5] to communicate in interaction scenes. Protocols use the FIPA language to build dialogs between agents. [FIPA] The FIPA language consists of a number of *communicational acts (CA)*. These CA can be used to recognize roles. We assume that the essence of a role (e.g. a police officer), which is the objective (e.g. catch thieves), remains the same. The FIPA expressions a role uses on the blackboard can be different in every organization. These differences in expression to define a role we call a *dialect*.

1.5 Organization of the Thesis

In the rest of this thesis we will do research whether an observed pattern of communicative acts can determine the role enacted by an agent.

[3] A blackboard is a place where agents communicate, like on a web-forum.
[4] Interaction scene; a pre-created scenario that is used to reach an corporate objective. (in the case when the multi-agent system represents an organization.)
[5] Protocols; are rules that tell the role-enacting agent how to use communicative acts to make a dialog.

The literature review provides an overview in what roles are defined and how they can be recognized. In here we also provide information about what communication is and what communicative elements can be used to recognize roles. Also we try to find some evidence if there are also roles in a social system, like an organization. We also will provide information about the recognition process in a human cognitive sense. After that we will propose two methods that might simulate these processes in a computational sense. This can be used for testing the hypotheses that FIPA communicational acts in dialogs can be used to recognize roles.

The section 3 about the method for role recognition provides a test case build upon the thesis scenario. Two classification methods are tested here, XML validation and Naive Bayesian classification. In section 3 a more detailed description of the scenario is given. This section 3 will also explain how a protocol will be used in practice. From this protocol dialect dialogs are made to test how well each classification method can find characteristics to recognize a role.

The last sections 4 and 5 are about the conclusion and discussion where we will discuss what method is best to use in practice. Also this section will provide the answer to the research question for this exploring thesis, along with recommendations for future research.

2 Literature Review

2.1 World of phenomena

"*No phenomenon is a phenomenon until it is an observed phenomenon*" – Niels Bohr

In the reality we experience we give names to events, objects and persons. According to Kant we only know or talk about the things we experience in our reality. Phenomenology is a branch in philosophy that starts with the things we experience (e.g. a role), and tries to extract the essential features of experiences or essence of what we can experience. Also a role, like a police officer, is a phenomenon. There are essential features about a police officer that makes this role what it fundamentally is. [wikipedia, phenomenon]

In this thesis it is explained that the essence of a role will be, among other properties it possesses, basically the objective it has to accomplish. (See section 2.4.2) For example the objective of a police officer is to catch thieves. This objective is often not explicitly observable. The general use of phenomenon is that it is an event that can be observed. The objective of a role cannot be observed unless the role-enacting agent creates observable events that can be experienced, like talking, making gestures, touching or producing a smell. [wikipedia, essence]

These events are captured and the essence is extracted to recognize this phenomenon the next time these events are experienced. For roles in a multi-agent system we focused on communication events to extract the essence. The communicative events have to be classified in attributes. The essence of a role will be the specific set of attributes that make the role what it fundamentally is. The attributes of language will be presented in the section below.

2.2 Communication

In this section the definition of communication will be provided and when can be used to identify roles. This investigates the third research task.

Communication: To communicate is, according to its Latin roots, "to make common", to make known within a group of people. People have to coordinate closely to make a piece of information common for them. [Clark, 1996]

If we want to perform joint activities like dancing, having a conversation or getting people to the moon we need to communicate. These joint activities are inseparable from using language. Joint activities are performed by participants through a variety of identifiable joint actions [Clark, 1996]. For example in conversations, in order to understand each other many of these joint acts are used. Each joint action contains a combination of a locutionary act and an illocutionary act. The locution contains the content of the joint action (to have a cookie) and the illocutionary act contains how the joint action should be interpreted (as a question).

At this point we will use the term "communicative act" in the remainder of the thesis in stead of the terms "illocutionary act". The term communicative act is used by FIPA to indicate the illocution of a message. These communicative acts are performed autonomously by the speaker and are identifiable. Communicative acts can be for example: a promise ("I'll be right back") or a request ("Can I have a cookie?").

In this thesis we hypothesize that a pattern of these communicative acts can predict what kind of role a person enacts. This is based on the fact that persons who enact a role can be recognized in the way they express themselves. When people use verbal communication, the way they express themselves will be in communicative acts.

2.2.1 Protocol

A protocol is a set of rules which guides how a joint activity should be performed [wiktionary, protocol].

Protocols are mostly used in non personal settings, like institutions. Participants in institutions perform communicative acts that resemble ordinary conversations, but are guided by institutional rules. Think for example of a lawyer interrogating a witness in court ([Clark, 1996] pp.5).

Protocols of such kind are also used in organization modeled multi-agent systems. In these organizations role-enacting agents use protocols to guide the agents through a conversation. We assume that each role prescribes a specific protocol an agent has to use to meet the role objective. Therefore we state that the use of protocol can describe the role.

2.2.2 Dialect

In our perspective we define the dialect differently than one normally has in mind.

Definition: dialect

Normally a dialect is a variation of an official language that is supported by institutions. [wikipedia, dialect]

This variation of an official language would appear in the locutionary part of a joint action. Some words are different, but the essence remains the same. For example in the Dutch official language the utterance: “Ik ga eten halen.” the essence (to get food) remains the same in the Limburgian dialect (used in the Dutch province of Limburg) when uttered: “Ich goan aete hoale.”.

In our perspective we consider something as a dialect when variations appear in the illocutionary part, where the essence also remains the same. When we speak of the essence, we narrow this definition by using the objectives of a role. The concept of the objective is discussed in the section about the role.

To show that the objective can be considered as the essence of a role the following example is provided. The objective of the role-enacting agent is to sell 200 boxes of Paracetamol. Two different types of communicative acts can be used to reach this objective. The first type is a “proposal” when providing the utterance: “*I offer you €15,- per box*”. The second type is a “request” when providing the utterance: “Can you s*end me €15,- for every box?*”. In both sentences the same objective can be achieved, which shows that the essence of what these sentences want to achieve remains the same. From this we can hypothesize that the essence of a role is able to remain the same by using different dialects.

2.3 Organization

2.3.1 Software Agents & Multi-agent systems

This section provides information for research task 2 by introducing the definitions of a software agent and a multi-agent system. Also information about the relationship between software agent, multi-agent system and a role will be provided.

Software agent

“An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.” [Wooldridge, 2002]

Multi-agent system (MAS)

Multi-agent environments extend single-agent architectures with an infrastructure for interaction and communication. Ideally, MAS exhibits characteristics typically to be open, to have no centralized designer, to contain autonomous, heterogeneous and distributed agents, and an MAS provides an infrastructure to specify communication and interaction protocols. [Dignum, 2004]

A multi-agent system is an environment where all separate agents perform joint actions to complete a joint activity. This joint activity is performed by two or more agents. Even an entire organization can perform a joint activity.

To show the importance of role recognition in a MAS Dignum (2004) says that coordination and communication are important issues of a MAS. The MAS architecture has to provide a mean for agents to find each other. And that this "coordination should emerge from the relations between agents rather than be imposed by the infrastructure, and as such does not employ centralized control but provides (mediation) services that facilitate the relations between agents." [Dignum, 2004]

2.3.2 Agent Societies

This section provides information for research task 2 by describing what a social society is and what the relationship of a role is in a society.

"The term *society* is used in a similar way in agent societies research as in human or ecological societies. The role of any society is to allow its members to coexist in a shared environment and pursue their respective roles in the presence and/or in cooperation with others. Main aspects in the definition of society are *purpose*, *structure*, *rules and norms*. Structure is determined by roles, interaction rules and communication language. Rules and norms describe the desirable behavior of members and are established and enforced by institutions that often have a legal standing and thus lend legitimacy and security to members. A further advantage of the organization-oriented view on designing multi agent systems is that it allows for heterogeneity of languages, applications and architectures during implementation." [Dignum, 2004]

2.3.3 Organizations and multi-agent systems

This section provides information for research task 2 by introducing the relationship between roles and organization.

Organization

"An organization is a group of people or other legal entities with an explicit purpose and written rules." [wikipedia, organization]

We can see in Illustration 3 that software agents are able to interact with each other by changing a certain portion or 'sphere' of the environment. This change will occur when they perform a joint action. We saw in section 2.2 about communication that joint actions can be a communicative act or a 'physical' action; like in a chess game one agent changes the environment by moving a chess piece. If agents change the environment together in a joint action, they are considered to have an organizational relationship.

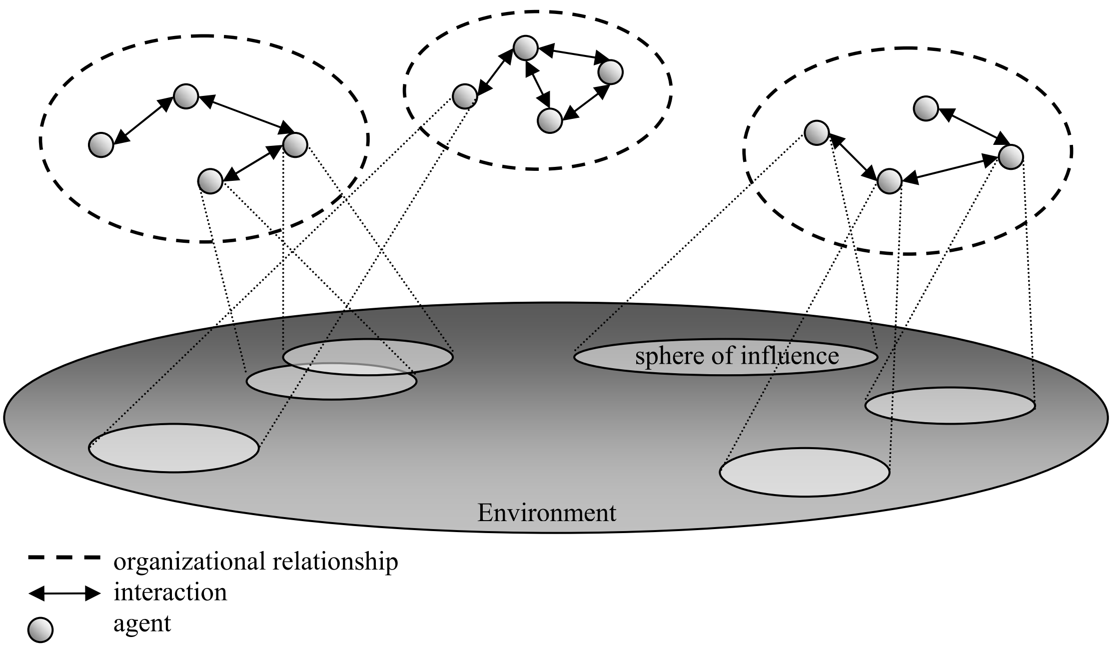


Illustration 3: Typical structure of a multi-agent system. [Jennings, 2000; cited by [Wooldridge, 2002], pp. 106]

In an organization agents work together in an organized way to deal with a complex task, yet each agent is designed to specialize in a certain simple task. For example: a medical team performing heart surgery. In this joint activity each team member performs specific joint actions with the other members of the team: one person monitors the scanner, one takes care of the instruments and one does the actual surgery. The joint activity is too complex for one person to achieve by himself.

Coordination is needed when the joint activity is split into several joint actions performed by agents. Roles are introduced, where each role has an objective and a mechanism how to act to meet this objective. Protocols provide the rules joint actions should be performed.

Illustration 4 provides an overview about the relationships between the agent, role, organization and environment; Odell and van Dyke made a social model in UML. [Odell, 2002] This model shows that software agents enact a role (e.g. a manager); that a role works in a group (e.g. financial department, a pharmaceutical company); that all roles are supported by an environment (e.g. the internet); that a role uses protocols and actions to interact and influence the environment.

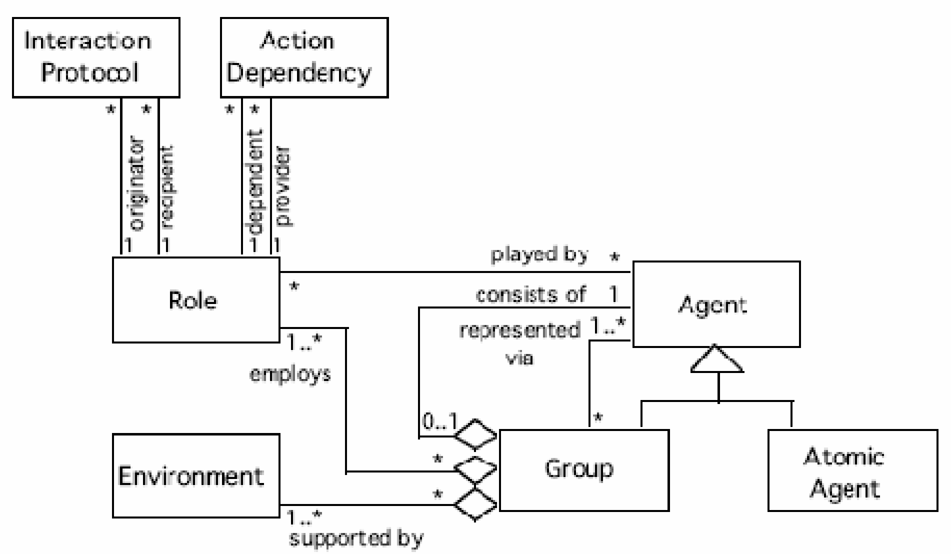


Illustration 4: Consolidated ontology; enhanced AALAADIN model in which social structures are represented in UML [Odell, 2002]

2.3.4 OperA agent organization model

This section provides an overview for research task 2 by introducing the OperA model. The OperA model from Dignum (2004) is a framework for describing a multi-agent system.

This framework helps us describing the aspects that are needed to create the thesis scenario in chapter 3. These aspects concern the place of a role in an organization and the communication between role-enacting agents. This framework shows us a direct link between the use of communicative acts and a role. Using this framework provides us a way to recognize a role by observing communicative acts.

The opera model envisions the multi-agent system as an organization of communicative socially engaged role-enacting agents. Here the agent is involved in a more human-realistic way of interaction. This interaction is built out of a complex network of properties from three different layers; the organizational layer, social layer and interaction layer, which are connected with each other.

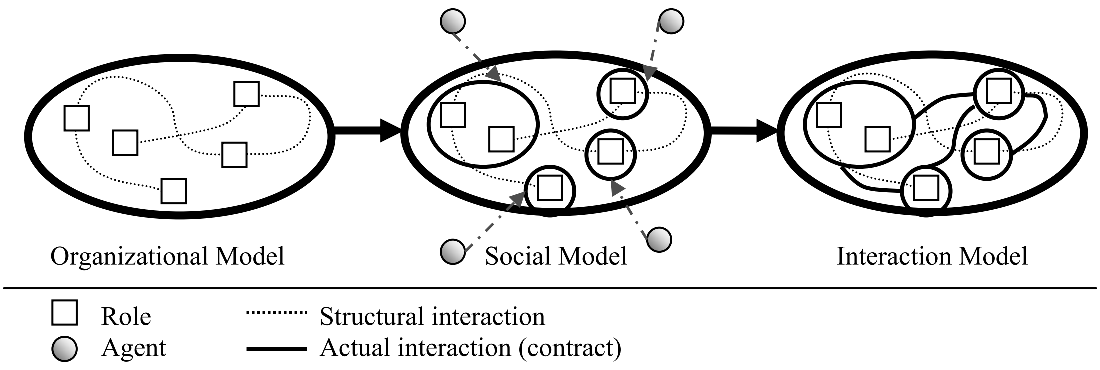


Illustration 5: organizational framework for agent societies. [Dignum, 2004] pp.53

As shown in Illustration 5 above; the organization operates in terms of roles. In the organizational model these roles in the organization fulfill a small task. The social model describes the relationships and capabilities of the roles. These individual agents have their personal objectives. The agent can enact the role if there is minimal conflict between the personal and role objectives. Also an agent can occupy more than one role in an organization. The interaction model describes what happens when role-enacting agents communicate. This means that it provides scene scripts that tell the agents what to achieve and what roles have to be involved in this scene to achieve a scene result.

The interaction model provides a list of scene scripts. These scripts describe the roles involved, the results that have to be achieved, and the desired states which provide the order of objectives to complete. The result of the scene is provided by the objectives that come naturally from the roles involved in the scene.

The basic idea is that these objectives roles have, can be achieved by agents using communicative acts in a particular way to create a situation that makes the agent able to achieve the role objective. Every role in the organization has a specific objective which has been translated by the agent to communicative acts. These communicative acts (CA) we are able to observe. From these CA we eventually recognize the role an agent enacts.

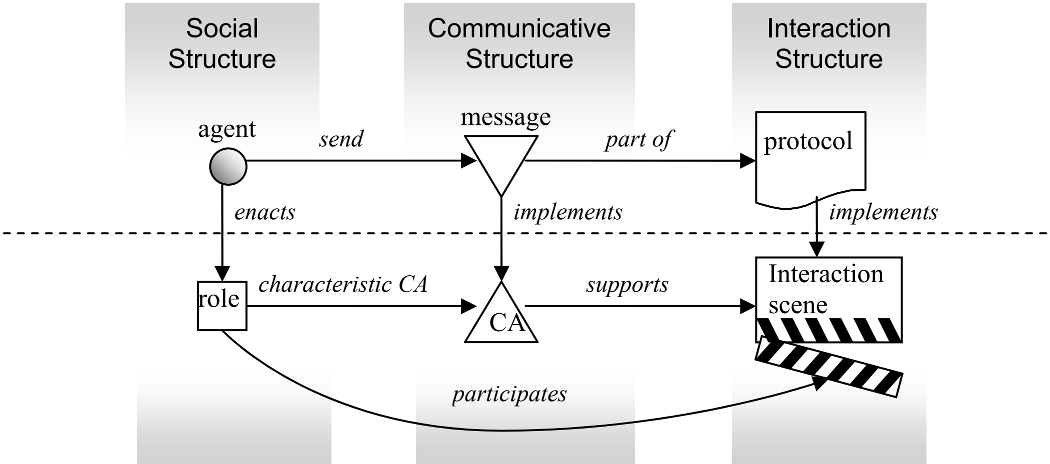


Illustration 6: Communicative Acts link roles to scenes. [Dignum, 2004] pp.75,92

The above Illustration 6 has been taken from the OperA thesis of [Dignum, 2004]). It shows that roles are defined in a social structure. Also it shows that these roles participate in an interaction scene. The roles in this interaction scene are using characteristic communicative acts.

The combination of a role and the protocol used in the interaction scene, generate very specific communicative acts. With these very specific communicative acts we can determine the perspective an agent has when he used the protocol, and eventually the role it has in the social structure.

2.4 Definitions of a Role

2.4.1 Human roles

Definition: role

In the human perspective Biddle describes roles as "*patterned human behaviors*" [Biddle, 1979]. This perspective confirms the characteristic use of communicative acts described in section 2.2 (and see section 2.4.4).

According to Biddle these behaviors are created by the consideration of several key concepts like *social position*, *expectations*, *context*, *functions*, *social system* and *socialization*.

Biddle tells us that when the behavior of an individual follows the same pattern over and over again, this person acts in a role with a certain social position, expectation, context, etc. This collection of fixed key concepts is considered as a role. This role can be identified by giving it a name.

For example the name of a role can be a traffic police officer. The person who enacts this role has a fixed social position (e.g. sergeant), fixed expectations (e.g. to uphold the law), fixed context (e.g. in traffic), fixed functions (e.g. arresting someone), fixed social system (e.g. the people at the police department and the people on the city streets), fixed socialization (e.g. to perform the correct actions to be successful)

2.4.2 Software agent roles

When talking about software agents, Wooldridge (2005)says that:

> "Roles are specified in terms of their responsibilities, permissions, protocols and activities."

The above quotation closely resembles the definition of [Dignum, 2004]. She describes the roles in touch with the environment as follows:

> "Role descriptions identify the activities necessary to achieve society objectives"

Dignum presents in Table 1 the elements that are used in the OperA model to define a role. We can see a resemblance in terminology also used by Biddle and Wooldridge.

Table 1: Elements of a role definition [Dignum, 2004] pp. 59

Role definition	
Role id:	A unique name which refers to this role
Objectives:	A set of landmarks that describe the desired results of this role
Sub-objectives:	A set of landmarks that describe desired intermediate states for role objectives.
Rights:	A set of expressions identifying the rights of this role
Norms:	A list of normative expressions that apply to this role
Type:	Indicates which type of agent can apply for this role, either internal or external.

The properties of a role are objectives that define the expectations what to achieve, norms that define the expectations how to behave, the rights that describe what a role is limited to, and the type of enactment that indicates if a role is originating internal or external form the organization.[Dignum, 2004]

Illustration 7 shows us that a role is defined by several concepts. To give an example we used the concepts provided by Biddle (1979) and Dignum (2004). These concepts we consider as internal parameters of an individual who performs a role. These internal parameters make the individual behave as it does. This individual uses ways to express his behavior which is on another level. This level is provided in the next section about expression features.

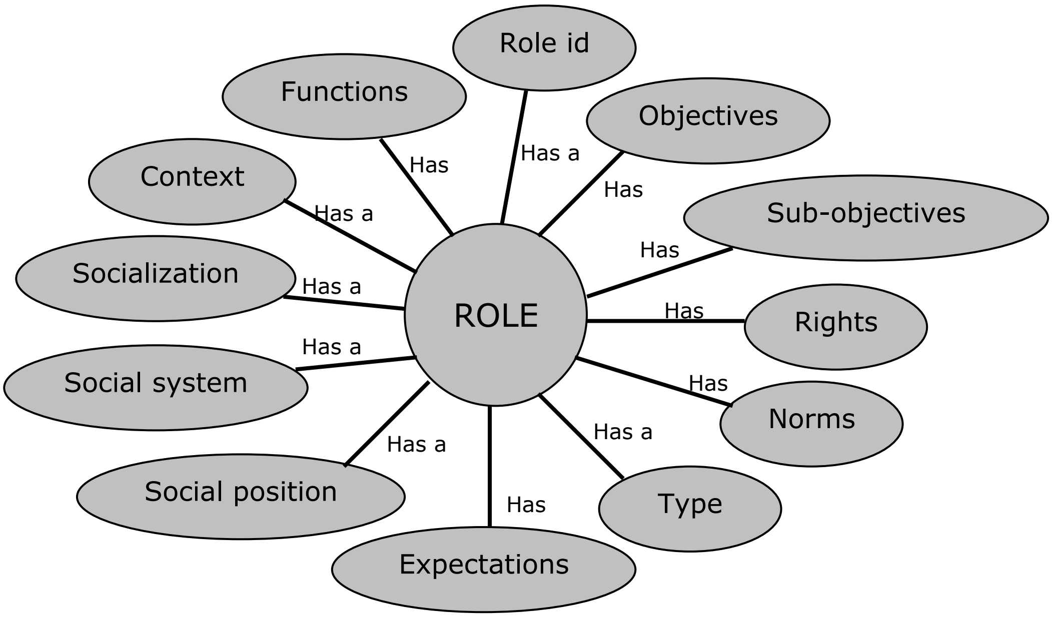


Illustration 7: Internal parameters that define a role

2.4.3 Expression features

An individual can express himself to another individual with verbal and nonverbal communication. One of these expression features can be communicative acts as we described in section 2.2. With the aid of theatre sciences we explore how humans express themselves. We zoom into the expression features of a software agent, and to the communicative acts we use in the thesis scenario.

Expression features of humans

Since humans can express themselves in numerous ways, the categories of these features are plentiful. However, with the help of theater science we will sum up a small number.

Within the field of theater Kowzan created a taxonomy that helps to classify the way of expression of the actor [Aston, 1991]. Table 2 shows us that an actor can express himself through the audio and visual channel by the means of the text he is saying, body language and how he looks like. The text he speaks can vary in words and tone, expressions of body language differ in mime, gesture and movement, and the way the actor express himself by his looks can vary in costume, make-up and hairstyle.

Table 2: Kowzan's classification of sign-systems [Aston, 1991]

Expressions	Transmitters	Location	Continuum	Channel
1. Word *2. Tone*	Spoken text	Actor	Time	Auditive
3. Mime *4. Gesture* *5. Movement*	Expression of the body	Actor	Space & time	Visual
6. Make-up *7. Hairstyle* *8. Costume*	Actor's external appearance	Actor	Space	Visual
9. Properties *10. Settings* *11. Lighting*	Appearance on the stage	Environment	Space & time	Visual
12. Music *13. Sound effects*	Inarticulate sounds	Environment	Time	Auditive

When the above table does not specify the expression features enough, even more features can be found in the book "The semiotics of theatre and drama" [Elam, 1980]. Table 3 below is used for stage directions by directors. In the book the table was bigger and extended to the fields of environment, technique and cinematography. In Table 3 we kept this close to the expressions an actor can perform.

Table 3: Stage directions give a perspective in human expression capabilities. [Elam, 1980]

Character: identification	Character: physical definition	Character: vocal definition	Speech: formal concerns
Description at first entrance	Entrance	Facial expression	Addressee: self
Detailed description at or prior to first entrance	Exit	Mode of delivery	Addressee: other
Occupation	Manner	Tone: quality of voice	Addressee: audience
Dominant trait(s)	Carriage	Tone: emotion	Aside
Relationship to others	Posture	Pace	Silence/pause
	Gesture	Volume	Song
	Movement	Rhythm	
	Action: self-directed	Mannerism	
	Action: other-directed	Emphasis	
	Action: self & object	Non-verbal	
	Action: other & object	Role-within-role	
	Reaction		
	Dumb show		

The tables above show how many features humans are capable of using to express themselves. From all these expressions the human brain is able to recognize the role that an individual enacts.

Illustration 8 shows us that a protocol is used for a specific role. This protocol is a set of rules (section 2.2.1) and tells the agent how to use the expression features. To give an example, the expressions in Table 2 are used. In the next section we provide examples of expression languages in computer software for intelligent agents.

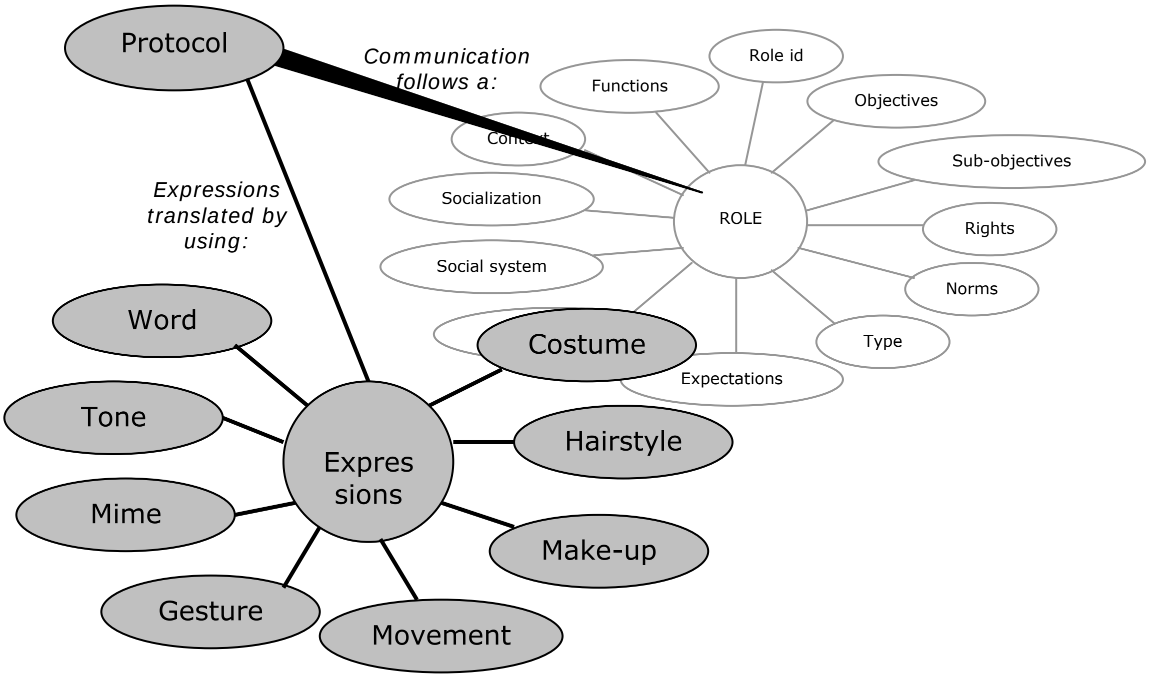


Illustration 8: External expressions that a role-enacting person uses to communicate with others

Expression features of software agents

In this section we show various examples of expression languages that have been developed for software agents. These languages are mostly written in XML. The expressions, like in Illustration 8, are used by software agents to communicate with humans or other software agents. Below we introduce two types of expression languages. The first is used for non-verbal communication between software agents and humans, like facial expressions. The second language is used for verbal communication between software agents, like the spoken word.

Non-verbal communication

For example; the virtual persons in a computer game, are called embodied agents. These agents perform facial expressions and body language.

Like HTML is a mark-up language for texts; several mark-up languages have been developed for embodied agents to mark-up their facial and body expressions. The markup languages for facial and body expressions that have been developed in parallel are called CML[6], HML[7]and VHML[8]. Each of the languages has the same purpose: to give expressions to embodied agents.

In Table 4 below an example has been given about the way VHML is used to express emotions in XML style. The VHML engine recognizes these tags and translates them to animations instructions for the embodied agent to a sad expression in face and body, to emphasize certain spoken words and to blink the eyelids in a certain point in the text.

Table 4: example of VHML [talkingheads]

```
<sad>
      you <emph> said </emph> to me once <pause length="short"/> that
      pathos left you unmoved, but that beauty <blink/> <emph affect="b"
      level="moderate"> mere </emph> beauty, could fill your eyes with
      tears.
</sad>
```

The above mark-up languages have a clear focus the nonverbal communication were emotions that are made explicit within a text. The above languages do not make explicit the rhetoric nature of the message. For example; if the text withholds a question, an answer, etcetera. [McTear, 2002]

[6] "Character Mark-up Language (CML) and is an XML-based character attribute definition and animation scripting language, designed to aid in the rapid incorporation of lifelike characters/agents into online applications or virtual reality worlds."[arafa, 2003]
[7] Human Markup Language, [HML]; follows the same description as CML.
[8] Virtual Human Mark-up Language, [VHML]; follows the same description as CML.

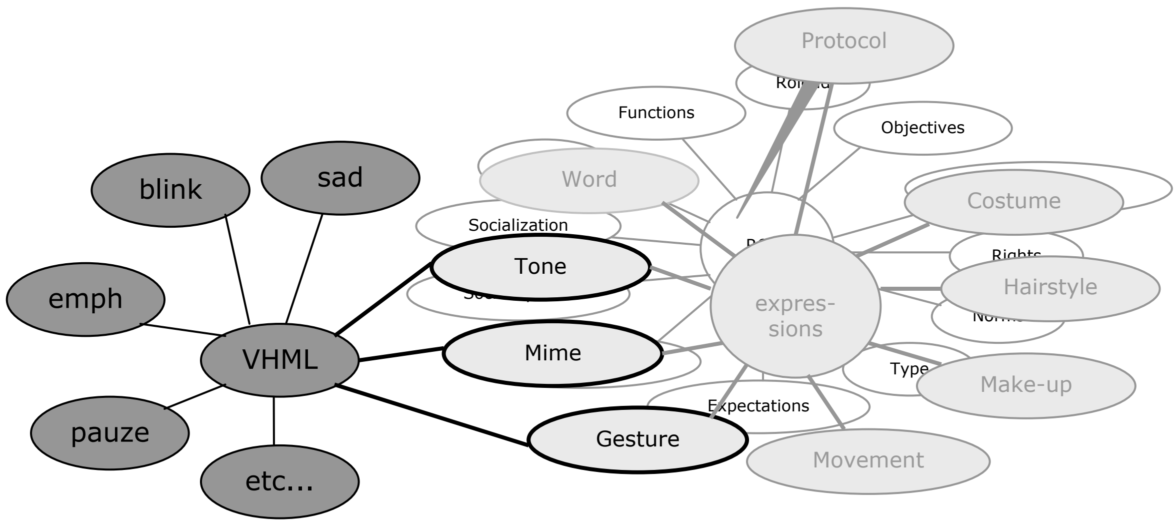


Illustration 9: VHML is used as an implementation of expression features.

Illustration 8 shows that VHML is used as an implementation to express the tone of voice, the mime and gesture of a software agent.

Verbal communication

For verbal communication between software agents we found FIPA ACL (Foundation for Intelligent Physical Agents Agent Communication Language) as a commonly used communication standard which provides the rhetorical meaning of a message by using communicative acts as an envelope. (Section 2.4.4 will tell more about protocols and communicative acts.) Table 5 shows us that the content of the message is enveloped in the communicative act "inform". The example, written in the standard FIPA mark-up, shows us that agent 1 wants to inform agent 2 that the price for "good2" is 150.

Table 5: Example of FIPA-ACL [Wooldridge, 2002] pp.175

```
(inform
      :sender     agent1
      :receiver   agent2
      :content    (price good2 150)
)
```

To provide a comparison with the nonverbal mark-up languages, the above example can be translated into XML according to FIPA specifications which are shown below. Unlike the mark-up tags that that describe emotions of the text used in VHML example, the mark-up of the message in Table 6 shows us the intention of the message by providing the communicative act.

Table 6: Communicative act are an envelope for the content in XML style. DTD used from http://www.fipa.org/specs/fipa00071/SC00071E.html

```
<fipa-message act="inform" sender="agent1" receiver="agent2">
      <content>price good2 150</content>
</fipa-message>
```

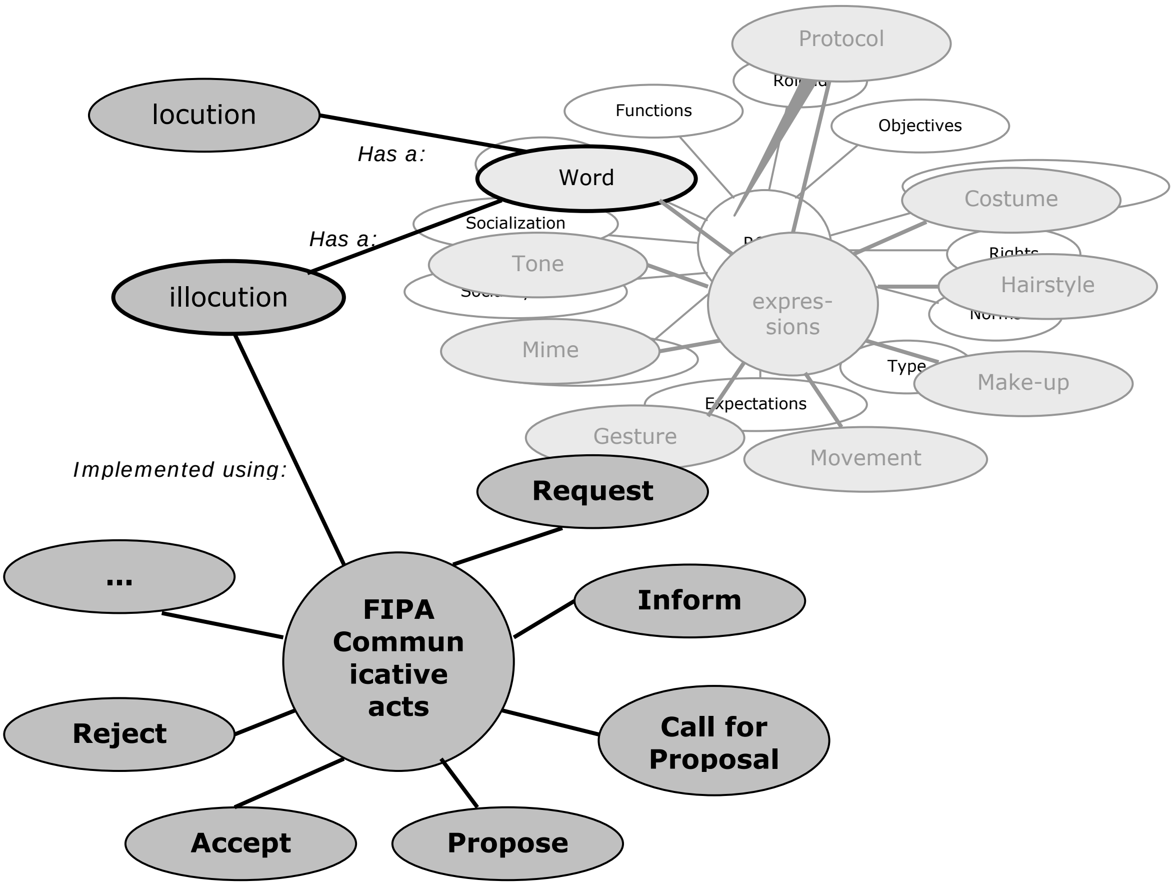


Illustration 10: The implementation of illocutions a software agent uses to express himself.

Illustration 10 shows us that FIPA communicative acts are used as an implementation to express the illocutions of words of a software agent. The FIPA communicative acts are part of the FIPA agent communication language (ACL). This language is used for communication between software agents.

Because we want to use the FIPA ACL, more information about this language will be provided in section 2.4.4 . This language will provide quantifiable attributes from which the essence of a role can be extracted that makes the role that it fundamentally is. These quantifiable attributes can be used in the recognition methods described later on in section 2.5.2.

2.4.4 Agent communication language

FIPA ACL is the part of the FIPA architecture that describes the Agent Communication Language. ACL is a language in which communicative acts are a part of messages. A FIPA ACL message consists of the several parameters such as a communicative act (see below), sender and receiver and a protocol. More parameters are provided in the appendix section 8.1.

A dialog between two agents consists of two or more messages. These messages follow a specific protocol that a role is obligated to use. Which protocol is used is stated in the protocol parameter.

The protocol parameter (which is also part of the FIPA message structure, see appendix section 8.1) is used as a method to provide control over the conversation. In the thesis scenario the protocol is used by agents in an organization which is specified in the interaction scene (see section 2.3.4). In the thesis scenario we made the assumption that a protocol in another organization can have the same name but works differently, or is named differently but work exactly the same (see section 2.2.2 about dialogs). More about the protocol and FIPA communicative acts will be explained in the following sub-sections.

FIPA communicative acts

The communicative acts are the actual joint acts we want to use to recognize the role of a role-enacting agent. FIPA specifies these acts in Communicative Act Library (FIPA CAL). Examples of communicative acts are:

accept-proposal, agree, cancel, cfp, confirm, disconfirm, failure, inform, not-understood, propose, query-if, query-ref, refuse, reject-proposal, request, request-when, request-whenever, subscribe, inform-if, inform-ref, proxy, propagate.

The FIPA-specification of the communicative act describes the way the agent should interpret the act. The specification of a communicative act can be found in appendix 8.2.

FIPA Protocols

Within FIPA-specifications, a protocol is a way to control the flow of a conversation. This means that it tells what communicative act follows the previous one. An example of the Contract-Net protocol is shown below in Illustration 11. The initiator and the participant send and receive messages that include the communicative act, the content and other message parameters.

The protocol describes that the initiator starts sending a message containing a call for proposal (cfp), the participant can reply to this message with either a refuse or a proposal, where the initiator can follow by rejecting or accepting the proposal and as a last turn the participant can reply with the communicative acts failure, inform-done or inform-result.

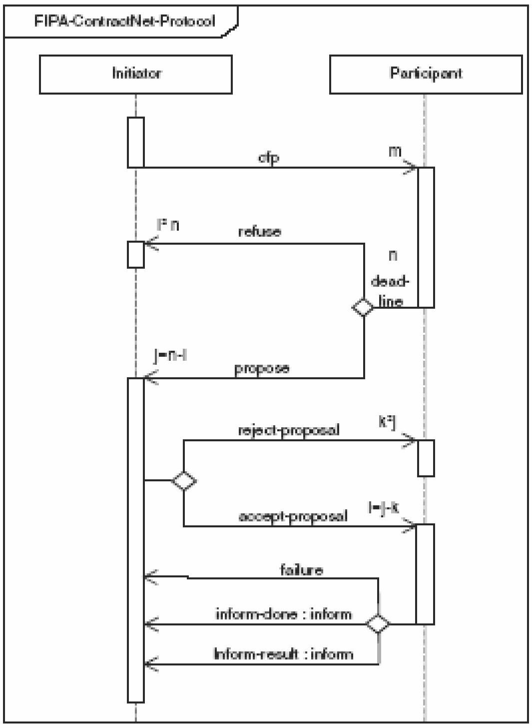


Illustration 11: Example of ContractNet-Interaction Protocol

As stated before; protocols used in multi agent systems are not an international standard, but custom made by an organization and they are not exchangeable with other organizations. When a custom protocol has the same name in one organization, it doesn't mean that the same communicative acts follow one another. Further research can be done for creating an international FIPA protocol repository.

Perspective

In Illustration 11 we can see two types of perspectives: the initiator and the participant perspective. A very important thing to notice is that an agent who uses a protocol can do this from either the initiator perspective or the participant perspective. In this research project these perspectives are call “protocol roles”.

Separating social and protocol role

At this point we will separate the protocol role an agent can enact and the social role an agent can enact. The social role will be the role in a social system, see section 2.3.4 about the OperA model. This can be for example the social role of a manager.

The social role of a manager uses a protocol role for communication in an interaction scene, for example the initiator role of the ContractNet protocol.

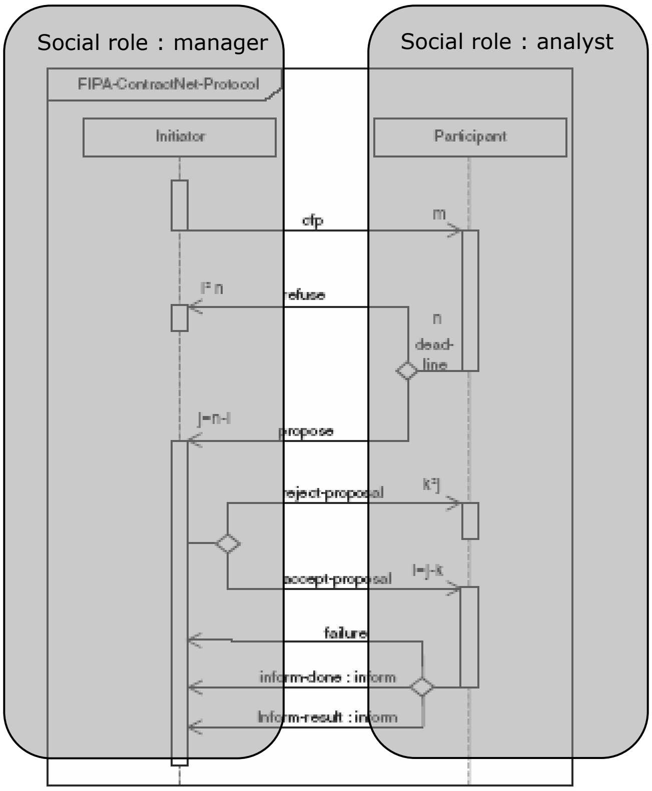


Illustration 12: Interaction scene where one social role always uses one perspective from the protocol to communicate. The manager uses the initiator side of the ContracNnet protocol, the analyst uses the participant side of the protocol.

Illustration 12 shows that the social role an agent enacts tells the agent what protocol role to use for communication. As we can see the protocol roles initiator and participant each use different communicative acts to be sent and received. These differences can be seen as typical characteristics of a protocol role. Because the social role and protocol role are linked, these typical communicative acts can be used to recognize a social role from a conversation.

To recognize a social role, the sent and received information of the communicative acts are used from the protocol role, a conversational perspective within the used protocol. This set of communicative acts will contain the essence of the role phenomenon. How to extract the essence and using it for recreation or recognition will be explained in the next section.

2.5 Recognition

In our research we will describe recognition as the process of recollecting patterns stored in memory by observing a given input [Anderson, 2000].

This section will be described how the human brain is able to recognize objects by describing the recognition process. Here we describe how the human brain is able to recognize the role of a role-enacting individual from the expressions this individual communicates. These human expressions we have seen earlier in section 2.4.3.

In a next step (section 2.5.2) computational techniques are studied that can be used by software agents for recognizing the role of a role-enacting agent.

2.5.1 Human Feature Analysis and Observation

Basically the recognition process is built out of two steps, learning and retrieval. The first step is learning how the observed input is represented. The second step is matching the new observed input if it resembles the earlier observed input.

Next follows two theories that support the recognition process, and a section about our interpretation about the recognition process in extend of the feature analysis theory.

Template Matching

The human brain is able to recognize the letter “A” in different handwritings because it has learned to categorize, store and recall the observed input signals of various channels (visual, auditive, haptic and scentive). Within our brain, the theory about *template matching* shows that we are able to imprint an observed input from a channel, categorize it and relate it to other imprinted memory to give it meaning. [Anderson, 2000] This imprint will be stored in long term memory after a learning phase. When we come across the same input that matches with the imprinted object, we are able to recollect it quite instantly.

However, there is a drawback to template matching that says we must have the capability to learn thousands templates of the letter “A” before we can read a different handwriting. Illustration 13 gives an example of the template matching theory illustrating photons fall on the retina.

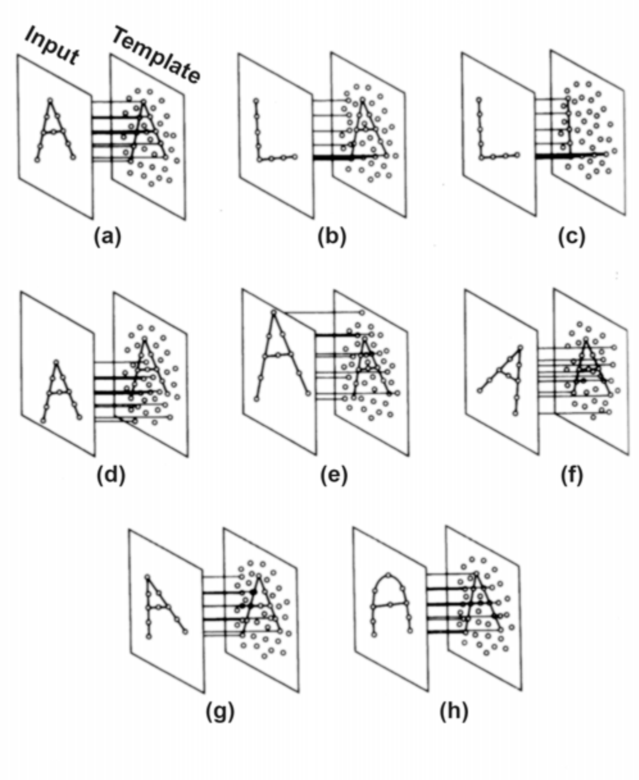


Illustration 13: Examples of template-matching attempts: (a) and (c) are successful attempts; (b), (d) to (h) failed. [Anderson, 2000]

Feature analysis

To overcome the drawback that the brain has to store thousands of representations of the letter “A”, theories about *feature analysis* and *object recognition* came across. [Anderson, 2000] These theories tell that the brain breaks down the letter “A” into unique atomic elements. For example this unique atomic element is a line, this line appears three times. /-\ . Every line is rotated and positioned differently. The features about the lines are stored in memory as a collection that represents the letter “A”. This collection can be seen as a pattern of features which can make a distinction between for example an “A” and a “B”

For objects, geometrics icons (sphere, cylinder, cone, etc) are the unique atomic elements. The mutual relations between them can represent the difference between a horse and a giraffe as shown in Illustration 14.

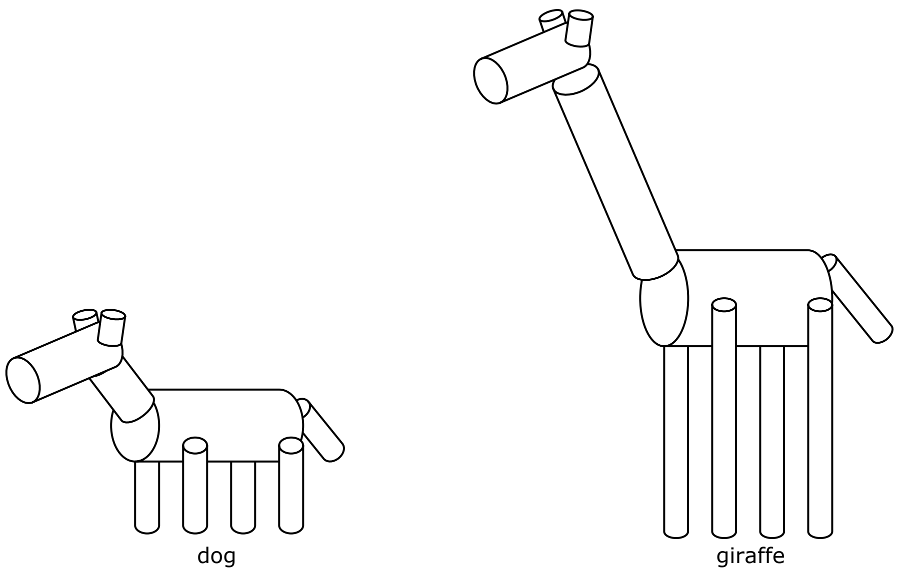


Illustration 14: Example of Object recognition which makes us able to differentiate between a dog and a giraffe using atomic elements, above geometric icons (geons). [Anderson, 2000]

Out of the combination of unique features we find a *pattern* which can represent a certain *concept* which has been learned in a learning process that is depicted in Illustration 15.

Pattern and Concept

To know what a pattern and a concept is, we need a definition. Let a pattern consists of atomic units that can be combined. And let a concept be a unique combination of atomic units, which will form the essence of a phenomenon. A quote about a pattern from Grenander (1993):

Pattern.

"The General Pattern Theory considers a pattern as a structure following three fundamental principles:

- Atomism: structures are constructed from units that are indivisible.
- Combinatory: create precise rules of deciding which combinations are allowed and which are not.
- Observability: given two combinations, when do they appear identical?"

[Grenander, 1993]

In the next section we use the term "concept". For better understanding we will provide a definition of this term. A quote about a concept from Rosengren (2000):

Concept

"To understand any phenomenon of the world we need to conceptualize it first. Conceptualization is a process of making distinctions along a dimension. For example the dimension of brightness we can distinguish what is light or dark. Names given to concepts are called *terms*.

When a given phenomenon can be characterized along more than one dimension we need to distinguish between more terms. Ordering terms to describe a phenomenon is called a typology. Typology is an instrument for classification. Human beings have been practicing it since the beginning of mankind." [Rosengren, 2000]

The above definitions are used to describe the recognition process in the following section.

Recognition process

Learning

In the recognition process we acknowledge that we learn patterns out of our environment and store them in long term memory as a certain concept. For example: we have learned that eating bread with green spots on it is not tasty. The concept "not tasty" is represented by the pattern of the units {bread + green spots}.[9]

Illustration 15 shows that we learn what a concept is if we see the same pattern of units or features[10] over and over again. ([Anderson, 2000] in: development of experience) This pattern is stored in long term memory along with the concept.

As we defined earlier a concept can be connected to other concepts. For example: "not tasty" can be connected to "sick" and "bad". When the pattern reappears with another observation

[9] Note that the unit bread and green spots are also both concepts that on their turn are patterns of other units. This can go on until we have units as small as the information about molecules, atoms, quarks or other atomic units we have not found out yet. For us communicative acts are the most atomic units to create patterns with. In computer terms we could go deeper; that communicative acts consist of words, letters, ASCII codes and binary signatures, but we don't go that deep.

this concept, along with all the connections to other concepts, is retrieved. ([Anderson, 2000] in: associative structure and retrieval)

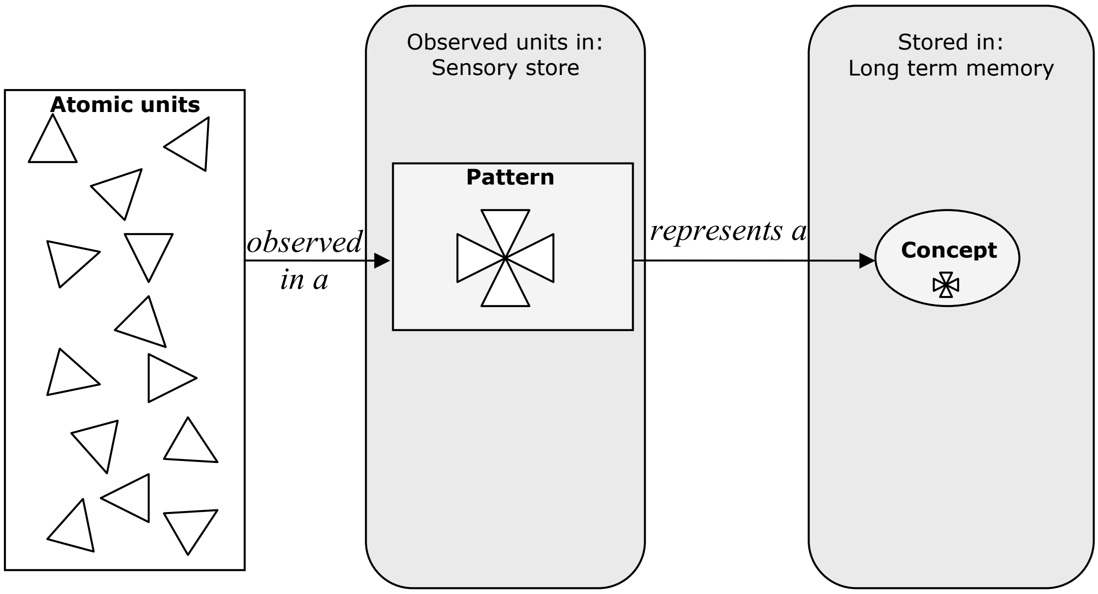


Illustration 15: A concept is represented by patterned atomic units. The units are observed by a sensor. The pattern is stored in sensory memory. And these patterns are linked to a concept and stored in long term memory. [Anderson, 2000](in: encoding and storage)

Retrieval and inference

Retrieval starts with an observation of a collection of atomic units. We want to match this collection of units to patterns we have seen before. In Illustration 16 we can see that observed signals are stored in sensory memory which resembles a pattern. This pattern is matched in the working memory, by using feature-analysis, to other patterns that are stored in long term memory.

Inference is a mechanism with makes us able to retrieve a target concept even if the patterns of atomic units is not complete or exactly the same. ([Anderson, 2000] in: retrieval and inference) The human brain fills in the missing link; which is also illustrated in Illustration 16.

[10] In this case the words features and units mean the same.

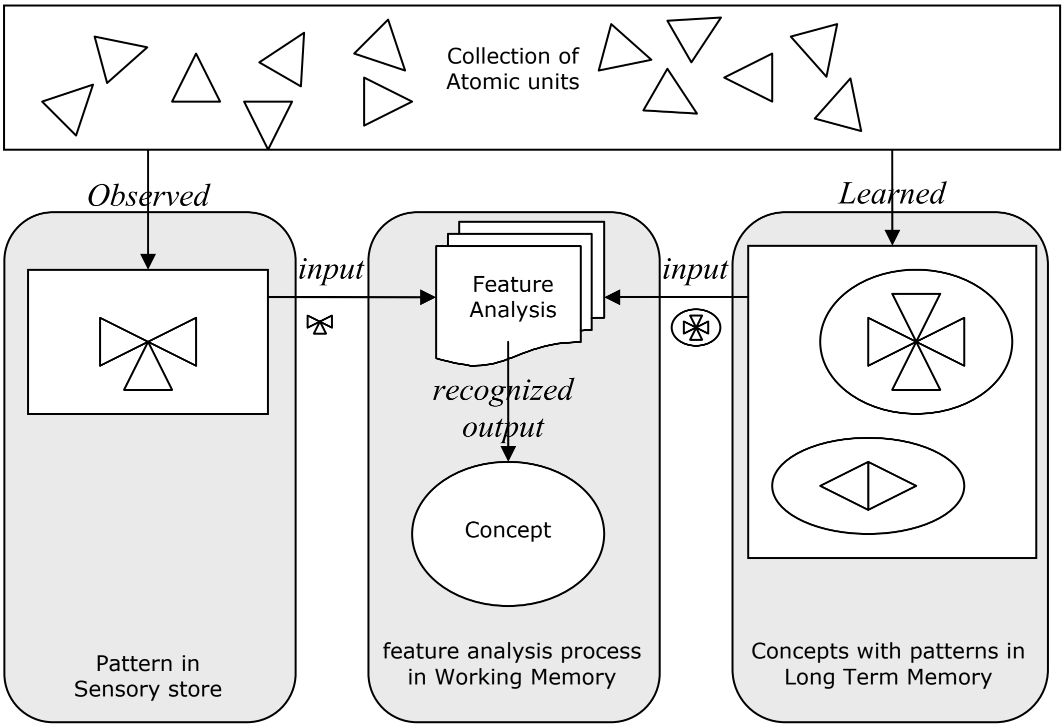


Illustration 16: Recognition by feature analysis of patterns

For example: To recognize a police officer we select the category ‘clothing’ from Table 2 to define atomic units for retrieval. The pattern of clothing units for a police officer can be a cap, a pistol, blue uniform and handcuffs. We infer to recognize police officers also from Germany who wear green uniforms instead of blue.

The example above shows that humans can recognize the role of a person even it does not completely match the learned pattern. These recognition problems also occur in the world of software agents when recognizing a role from a conversation. Methods for matching incomplete patterns will be reviewed in the next section.

2.5.2 Software Agent Feature Analysis and Observation

In this section we will explore two computational methods for recognizing roles using the XML validation method and classification with Naive Bayesian networks.

These methods follow the description of respectively deductive and inductive reasoning. Reasoning refers to the process by which new knowledge can be inferred from what already is known. [Anderson, 2000]

Deductive Reasoning with XML Strict Validation method

Deductive reasoning

Deductive inference leads to a form of reasoning that is mathematically exact. This means that if the premises are true, then the conclusion is guaranteed also to be true. [Darlington, 2000], [Anderson, 2000]

The method used to validate XML documents can be used for deductive reasoning. With XML validation we can recognize valid (true) and invalid (false) documents. Using XML gives us the ability to structure documents and validate it to a Document Type Definition (DTD). A DTD tells humans and computers how to structure a document, this way another person or computer with the same DTD can understand and read the document easily and automatically.

For example we use the following XML document to validate.

```
<A>
      <B/>
</A>
```

Then we have two DTD's that resemble a class.

Class 1:

```
DTD: A,(B|C)
```

Which says that element A occurs first, the next one can be either B or C

Class 2:

```
DTD: B,A
```

Which says that element B occurs first, the next to occur is element A.

Class 2 cannot validate the document, Class 1 can, even if element C does not appear in the document.

This validation method preserves the structural integrity of a document. As stated before, the conversations software agents have in a dialog follow a structure enforced by a protocol. The dialog can be transcribed into an XML document. (see section 3.4.2)

In the learning process, the agent who enacts role 1 is able to generate his own DTD from the

XML document of the conversations he has with role 2. To recognize a role-enacting agent in another organization that uses similar conversational structures the observer agent (who currently enacts role 1) only has to apply his DTD to the conversations he observes from dialogs of the other organization.

A lack of this method is that the classification for a conversation has only two possibilities: the document is valid or not valid. Yet there is a grey area when noise appears in the conversation, or the conversation is written in a dialect. To overcome this strictness we in a conversation, flexibility has to be created.

Inductive Reasoning with Bayesian Inference

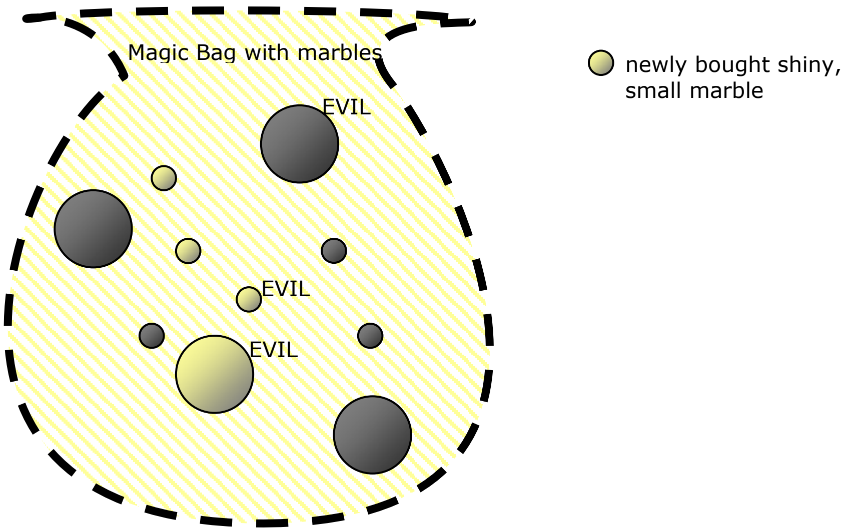
Definition: Inductive reasoning

Inductive reasoning is concerned with conclusions that probabilistically follow from their premises. This term is used to describe the process by which one comes to conclusions that are probable rather than certain. Mathematicians and philosophers have developed a model called "Bayes's theorem" that should reason in inductive situations. [Anderson, 2000]

An inductive situation can be to filter span from an e-mail box. Naive Bayesian Classification Networks are known for their flexibility to filter this spam from the e-mail box.

This section will describe the Naive Bayesian Classification (NBC) method. NBC and Bayesian Classification (BC) are both classification techniques. The only difference between NBC and BC is that NBC assumes all input variables to be independent from one another. Both techniques are able to learn and predict a class. The Bayes classifier is able to learn a class V from a tuple of attribute values $<x_1, x_2 \ldots x_n>$ where subsequently the learner is asked to predict the class of a new tuple. [Mitchell, 1997]

The two approaches will be explained with an example. In this example we have a bag with marbles, and we know quite a few things of the content. It has 10 marbles, which have attributes variables in size and coating: four have a big size and six small, four have a shiny coating and six not. And from this collection seven are classified as "Good" and three are "Evil".



In the example we have a new target: We just bought a new **shiny**, **small** marble from the magic store. When we put it in the bag the marble magically turns to "Good" or "Evil". The aim we have is to predict which class (Good or Evil) this marble will be in.

The Bayesian network

The Bayesian network tells us how to classify the marble we picked from the bag by assigning the most probable class ("Good" or "Evil"), v_{MAP}, given the attribute values (size and coating) that describe the marble. The expression looks like:

Equation 1: Bayesian Classifier Algorithm [Mitchell, 1997]

$$v_{MAP} = \underset{v_j \in V}{\text{argmax}}\ \mathrm{P}(v_j \mid x_1, x_2 \ldots x_n)$$

Which can be rewritten as:

$$v_{MAP} = \underset{v_j \in V}{\text{argmax}}\ \mathrm{P}(x_1, x_2 \ldots x_n \mid v_j)\mathrm{P}(v_j)$$

Where v_{MAP} is the class found with the Bayesian network, v_j is a class from class collection V, P is the probability of a class v_j, x_i are the iterated attributes, i is the iterator, and argmax is the operator that selects the outcome with the highest probability.

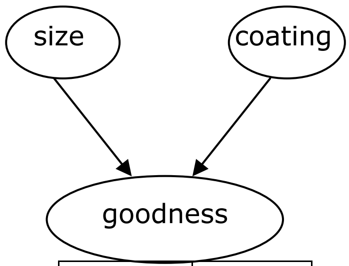
In the example this formula will look for the "Evil" class like:

$$\mathrm{P}(\mathrm{Evil} \mid \neg\mathrm{Big} \wedge \mathrm{Shiny}) = \mathrm{P}(\neg\mathrm{Big} \wedge \mathrm{Shiny} \mid \mathrm{Evil}) \times \mathrm{P}(\mathrm{Evil})$$

In here we can see that the combination of both attributes plays a role. In words it says: To know the chance that a ¬Big and Shiny marble is Evil we need to calculate the product of the chance that Evil presents itself as a ¬Big and Shiny marble *times* the chance Evil presents itself.

In the table below we continue with the example for the Baysian approach. The table shows us the combination of attributes when a marble is “Evil” or not. The classification for “Evil” is visualized below, where we look at the bag of marbles and we count for each given class of “Evil” how many times Evil is presented in the combination of the attributes Big, Small, Shiny or not Shiny:

Size		Coating	Tuples	S∧C |¬Evil	S∧C | Evil	P(S∧C | ¬Evil)	P(S∧C | Evil)
¬Big	∧	¬Shiny	3	3	0	3/3	0
¬Big	∧	Shiny	3	2	1	2/3	1/3
Big	∧	¬Shiny	3	2	1	2/3	1/3
Big	∧	Shiny	1	0	1	0	1/1



P(¬Evil)	P(Evil)
7/10	3/10

Illustration 17: Visualization of the Bayesinan network according to [Hugin]; for P(goodness|size∧coating)

In the Bayesian approach we calculate all combinations of all attributes with all possible values at forehand and look at the class with the highest outcome.

To find out if the small shiny marble is “Evil” we make the following equation:
P(Evil | ¬Big ∧ Shiny) = P(¬Big ∧ Shiny | Evil) × P(Evil)
= 1/3 × 3/10 = 3/30

To find out if the small shiny marble is “Good” we make the following equation:
P(¬Evil | ¬Big ∧ Shiny) = P(¬Big ∧ Shiny | ¬Evil) × P(¬Evil)
= 2/3 × 7/10 = 14/30

The operator argmax (P=14/30) tells us that the new marble will be classified as “Good”. There is no threshold in which the argmax operator considers the classification as a valid one. For example the threshold of all probabilities above 60% are considered as valid ones, is not a function of the argmax operator.

The Naive Bayesian network

The Naive Bayesian network also is able to classify the marble you picked from the bag by assigning it to the most probable class (“Good” or “Evil”), v_{MAP}, given the attribute values (size and coating) that describe the marble. The only difference lies in the way of calculation where the assumption for Naive Bayesian classification is that all variables are independent of each other! This means we do not need to calculate all combinations of all variables.

In the above approach the number of combinations increases when the number of attributes n increases exponentially with 2^n. The Naive Bayesian network philosophy is to simplify the process. It does this by taking the product of all individual attributes x_i for a certain class which is substituted into the Bayesian formula which gives the following:

Equation 2: Naive Bayesian Classifier Algorithm [Mitchell, 1997]

$$V_{NB} = \underset{v_j \in V}{\text{argmax}}\ P(v_j) \prod_i P(x_i \mid v_j)$$

Where V_{NB} is the class found with the Naive Bayesian network, v_j is a class from class collection V, P is the probability of a class v_j, x_i are the iterated attributes, i is the iterator, Π is the product mechanism, and argmax is the operator that selects the outcome with the highest probability.

In the example this formula will look for the "Evil" class like:

P(Evil | ¬Big ∧ Shiny) = P(Evil) × P(¬Big | Evil) × P(Shiny | Evil)

Here we can see that the individual attributes play a role. In words it says: To know the chance that a ¬Big and Shiny marble is Evil, we need to calculate the product of the chance that Evil presents itself as a ¬Big marble *times* the chance when Evil presents itself as a Shiny marble *times* the chance Evil presents itself.

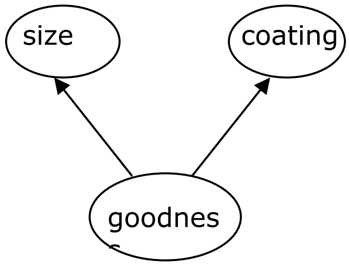
In the table below we continue with the example but now for the NB approach. The table shows us not the combination of attributes when a marble is "Evil", but the single attribute. We look at the bag with marbles and we count for each given class of "Evil" how many of them are Big, Small, Shiny or not Shiny which results in the following table:

	Attribute	***Number of:***		***Probabilities***	
	x_i	x_i | ¬Evil	x_i | Evil	P(x_i | ¬Evil)	P(x_i | Evil)
Size	¬Big	5	1	5/7	1/3
	Big	2	2	2/7	2/3
Coating	¬Shiny	5	1	5/7	1/3
	Shiny	2	2	2/7	2/3

In visualization perspective we get the following:

	P(Size | ¬Evil)	P(Size | Evil)
¬Big	5/7	1/3
Big	2/7	2/3

	P(Coating|¬Evil)	P(Coating | Evil)
¬Shiny	5/7	1/3
Shiny	2/7	2/3



P(¬Evil)	P(Evil)
7/10	3/10

Illustration 18: Modified Visualization for Naive Bayesian classification; for P(size | goodness) and P(coating|goodness)

To find out if the small shiny marble is "Evil" we make the following equation:
$P(Evil \mid \neg Big \wedge Shiny) = P(Evil) \times P(\neg Big \mid Evil) \times P(Shiny \mid Evil)$
$3/10 \times 1/3 \times 2/3 = 2/30$ (= 7/105)

To find out if the small shiny marble is "Good" we make the following equation:
$P(\neg Evil \mid \neg Big \wedge Shiny) = P(\neg Evil\ Evil) \times P(\neg Big \mid \neg Evil) \times P(Shiny \mid \neg Evil)$
$7/10 \times 5/7 \times 2/7 = 1/7$ (= 15/105)

The argmax operator tells us to choose from the class with the highest outcome (P=15/105): Also here the newly bought marble will be classified as "Good".
Again: There is no threshold in the probability output which the argmax operator considers the classification as a valid one.

Naive Bayesian network and it uses

In practical use the NB classifier is useful for classifying text. Text classifiers are used for learning which news articles one would like to read, or what web pages a person is interested in on the internet, or how spam is usually defined in e-mails.

The NB classifier is good at seeing through holes (missing values) and noise (scrambled values). It handles missing values very well because there is a relationship in a document between two words [Dunham, 2003]. For example with the words "keeper" and "soccer", which are strongly classified for the class "sports article", it doesn't matter which word is missing because the attributes are independently related to "sports article", not the combination of the two words. Also an uncommon word such as "grasshopper" will have little influence on classifying the article as a "sports article" as long as stronger related words remain in the document.

This classification method can also be useful in conversations which are typically specific for a role. The tuple of attributes can here be the verbal elements from FIPA-ACL, and the Role as a Class. The role can be found even when one is jumping in the middle of a conversation, or the conversation is being interrupted by small-talk, or told in a dialect where only portions of a conversation can be understood.

2.6 Summary

In the introduction in section 1.2 we have explained why role recognition is important. In the literature review is explained that agents in a multi-agent system have to enact roles to position itself in a social system. Detailed information is provided in section 2.3 which concludes research task 2.

The conclusion of research task 1 is that a role is defined by several internal properties. These properties form the essence of the role and make the enacting agent unique in its function in the multi-agent system. This essence for a role remains the same despite the way communication languages are used. Detailed information about these subjects can be found in sections 2.1 and 2.4

Also in the literature review is shown that one way of expressing these internal properties is by means of communication. Software agents communicate with each other in a multi-agent system using the FIPA communication language. The communicative acts from this FIPA language are used to transform ordinary conversations into quantifiable communication patterns. This paragraph concludes research task 3. Detailed information can be found in sections 2.2 and 2.4.4.

The communication patterns are used to recognize a role. This recognition is done by two different methods of deductive and inductive reasoning. XML validation is used as a method for deductive reasoning. For inductive reasoning the Naive Bayesian Classifier method is used. In both methods the communicative acts uttered by a role-enacting agent are stored in a pattern. This stored pattern is used to recognize role-enacting agents who utter the same communicative acts. The deductive reasoning method recognizes upon an exact match in patterns, and the inductive reasoning delivers recognition with a probability. This paragraph concludes research tasks 4 and 5. Detailed information can be found in section 2.5.

In section 3 we continue with the specification and implementation of the proposed recognition methods. To test the recognition methods, two data sets have to be made. One training set, for the method to learn the pattern to recognize the role. And a testing set to test what method scores better. In the leaning phase one interaction protocol is used to create a set of dialogs. In the testing phase one dialog from the learning phase is used to create variations in the set of communicative acts. These variations are made to simulate dialects (see section 2.2.2) from other multi-agent systems. Despite of these variations, the essence (section 2.1) of a role (section 2.4.2) has to remain the same. The objective a role-enacting agent has to achieve is a property that is considered to be essential (among other properties) for the definition of a role. To test the recognition methods, only those variations are used where the essence of the new dialog is equal to the essence that lies in the originating dialog. This forms the basis for upcoming section 3.

3 Methods for Role Recognition

In this section we will present a test environment of a multi-agent system that is built conform to the thesis scenario. This will create a platform that can be used for the computational reasoning methods. In the subsections we will explain why and how we used a type of computational method in order to classify roles from the given example.

In order to answer the research question written in the introduction we create a test environment of a multi-agent system that can be used to test the two reasoning methods described in section 2.5.2. To compare both methods, we create the following tasks in extend to research task 6.

1. Creating a test environment.
2. Creating a formal specification generally describing the thesis scenario and the test environment.
3. Creating test conversations.
4. Concerning deductive reasoning
    a. Creating a formal specification of the reasoning methods.
    b. Implementation of the specification.
5. Concerning inductive reasoning
    a. Creating a formal specification of the reasoning methods.
    b. Implementation of the specification.

3.1 The test environment

The test environment is based upon a framework that has been borrowed from the OperA model. In this environment we have not one, but two organizations, a pharmaceutical organization and an insurance company.

Roles & Role types

Each organization contains both role types: internal and external (see section 2.3.4). The insurance company contains internal roles with the Role IDs r1, r2, r3 and r4. The Role ID r2 is known in the insurance company as a Receptionist and Role ID r4 as a Manager.

The pharmaceutical company contains internal roles with the Role IDs r5, r6, r7 and r8. The Role ID r6 is known in the pharmaceutical company as a Receptionist and Role ID r8 as a Manager.

Also the pharmaceutical company provides an external role type with Role ID ω. This role is known as the observer. This observer has the right to observe the dialogs between the role-enacting agents at the company blackboard.

Protocols

The roles use one FIPA Interaction Protocol (IP) for each interaction scene. The communication lines in Illustration 19 represent the interaction scenes we introduced earlier in section 2.3.4. In this test environment we assume that Role r3 and r4 both are involved in one interaction scene and use the ContractNet IP to communicate.

Communicating

In this example the role-enacting agent r4 talks to role-enacting agent r3 about the new drug policy. The agent acting as r4 wants to make a deal with r3, but the objectives cannot be met. This conversation in FIPA ACL using the ContractNet IP might look like this:

```
r4: call_for_proposal (cfp)
r3: propose
r4: reject-proposal
```

R4 from the insurance company wants to meet his objective soon and crosses the border of his organization to another organization.

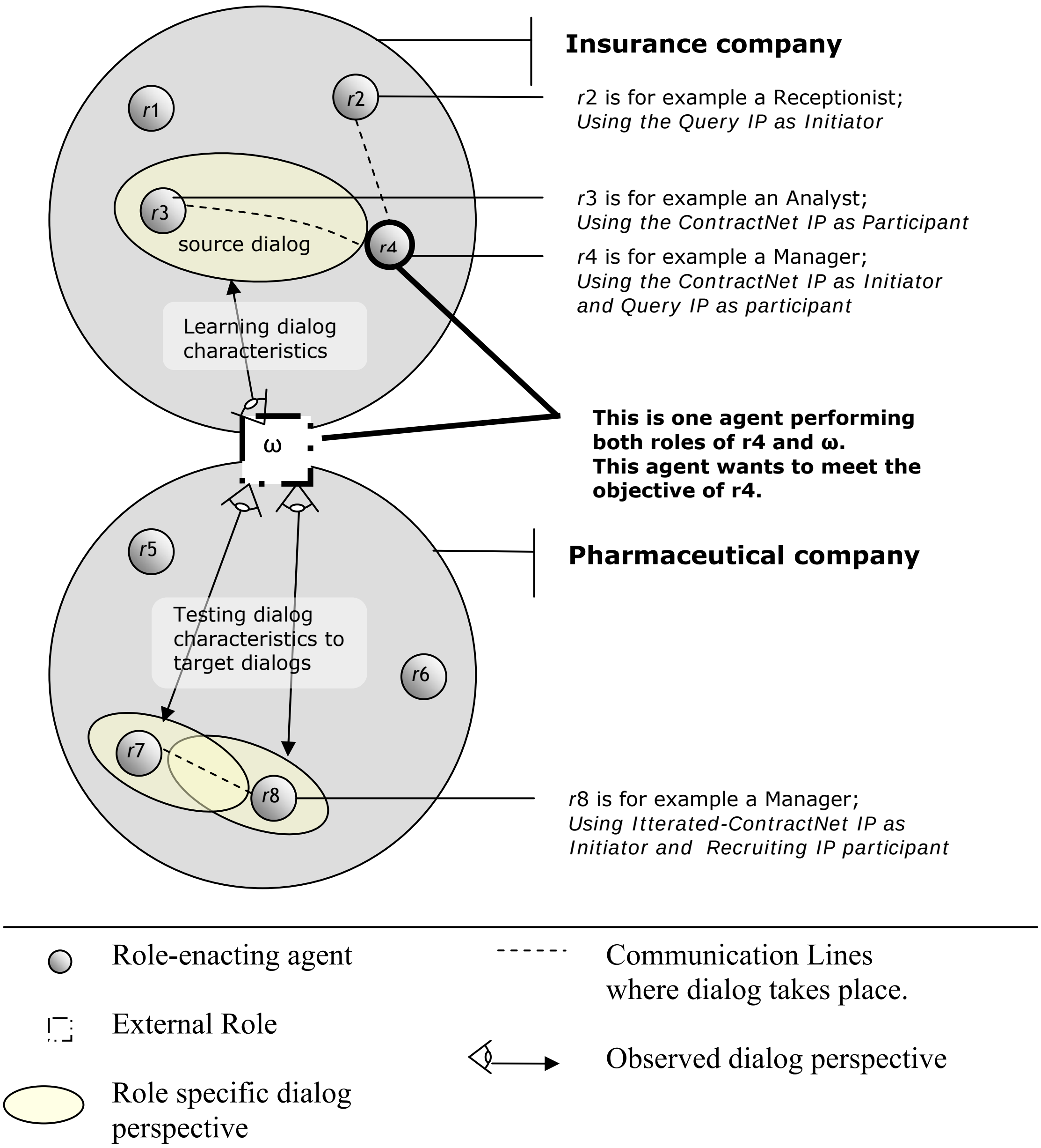


Illustration 19: Multi-agent system Test environment reflecting the thesis scenario. The observer learns dialog characteristics from role r3. Then he observes dialogs from the perspectives of role r8 and r7 to find similar characteristics. Role ω and r4 are operated by the same agent.

Observation

The agent who formally acted as r4 now occupies the role of observer ω in the pharmaceutical company.

In this test environment, ω observes the dialogs between role-enacting agents *r*8 and *r*7 and *r*8 and *r*5, which gives the following result:

Dialog between r8 and r7	*Dialog between r8 and r5*
r8 cfp r7: propose r8: reject-proposal r8: cfp r7: propose r8: accept-proposal r7: inform-done	r8: proxy r5: agree r5: inform-done-proxy

In the dialog between r8 and r7 we find similarities between the dialog *r*4 had with *r*3. The reasoning process confirming that they are similar can be found in sections 3.4 and 3.5.

Observer Objective

The objective of the role ω is to find roles that perform similar behavior in dialogs as r3. For this role we test the recognition ability by conducting deductive and inductive reasoning processes. (see section 2.5.2) For both reasoning processes, the observer has to learn the characteristics of an internal role; create a hypothesis and test this hypothesis on the internal roles of the other organizations.

Learning phase

In the learning phase ω observes dialogs in the perspective of the source role from the insurance company. This source role is in this case r3, the conversation partner of r4. The observed information consists of the sequence of messages in a dialog. And each message contains a FIPA communicative act with the information whether it has been sent or received. From this information a behavior pattern is created according to the different reasoning methods (this will be further explained in the implementation sections). This learned knowledge about r3 will be used to recognize another role that performs similar behavior to r3 in the pharmaceutical company.

Testing phase

In the testing phase ω observes dialogs of the target role α from the pharmaceutical company. Then ω uses the deductive or inductive reasoning method to proof that α is similar to the source role. In this test environment we let ω observe r8 and r7.

Test a reasoning method

To test the differentiation ability of the reasoning methods, both roles, r8 and r7, are considered a target role α. This provides conversations from two perspectives. This way we proof that the reasoning methods can tell the difference between one role and the other.

Also to test the flexibility of the reasoning methods we created six different conversations. The conversations will test the flexibility of the methods to handle dialects (see section 2.2.2). Dialects will be simulated by making variations in communicative acts from the source dialogs to the target dialogs. The source dialogs, we use in the learning phase, are created from the ContractNet IP (see Illustration 19). From one source dialog we created the six variations by, adding and replacing communicative acts. We will provide more information about the test conversations in section 3.3.

3.2 Specification of the test environment

To formalize the test environment, we start with the following assumptions:

- α is the target agent that is overheard
- ω is the observer agent that overhears a dialog δ^O
- δ^O is a dialog overheard in a conversation with α.
- δ is a dialog.
- ω has the ability to learn the dialog characteristics of the roles $r_1 \ldots r_n$ in his own organization.
- r_i is a role for which:
  - π^I_{ri} is the set of protocols this role utilizes in the perspective as an Initiator.
  - π^P_{ri} is the set of protocols this role utilizes in the perspective as a Participant.
- In the learning phase we want to classify a δ to a r_i.
- In the testing phase we want to classify a δ^O to a r_i.

We would like to specify for each role which protocols it uses in the perspective of Initiator or Participant. Making this distinction characterizes the use of communicative acts in a dialog.

For example: the agent who enacts the role of analyst r_{analyst} uses the Request protocol as initiator, the ContractNet protocol as participant and the Query protocol as participant. For π^{P}_{analyst} = {query, contractnet} and π^{I}_{analyst} = {request}

Observing a set of dialogs

The perspective differentiation delivers for the role a characteristic set of dialogs. The perspective of a dialog is defined by whether a communicative act is either sent or received.

To continue to specify the dialogs used by a role enacting agent we use the following:

- Δ_r is the set of dialogs δ permitted by π^{I}_{ri} and π^{P}_{ri}. $\Delta_r = \{\delta_1 \ldots \delta_m\}$
    - δ is a list of transmitted messages M. $\delta = <M_1 \ldots M_k>$
        - M is a tuple containing a communicative act and transition information. $M = <\lambda, T>$
            - λ is the communicative act
            - T is set to either *sent* or *received*

We continue from the previous example where the analyst uses the three protocols from a different perspective. Δ_{analyst} contains the dialogs of these perspectives. For the analyst one contractnet dialog looks like this:

```
<
        <cfp, received>,
        <propose, sent>,
        <accept-proposal, received>,
        <inform-done, sent>
>
```

For the analyst’s conversation partner, the manager, who also uses the ContractNet protocol this sent and received information is exactly the opposite.

```
<
```

```
        <cfp, sent>,
        <propose, received >,
        <accept-proposal, sent>,
        <inform-done, received >
>
```

Both dialogs define a perspective of the protocol which is characteristic for a role.

A role is defined by the dialogs from a perspective that are permitted by the protocol it uses.

$r \in \Delta_r$

3.3 The test conversations

The test conversations are possible dialogs r8 and r7 could say to each other. These conversations are used in the implementation to test the flexibility of each reasoning method. When implementing, these dialogs are transformed to a format for each reasoning method.

- The first dialog between *r*8 and *r*7 will be an original conversation from the source dialog of *r*4 and *r*3.
- The second dialog will simulate a dialect by having a loop in the middle of the conversation. This dialog is created by using the Iterated ContractNet IP, which is a FIPA standard.
- The third dialog we insert another dialog, created from the Request IP, into the middle of the original conversation.
- The fourth, fifth and sixth dialogs we create noise by replacing, adding and removing communicative acts from the originating dialog.

Below the dialogs have three columns. The first column is telling which agent is performing a communicative act, the second column is the communicative act and the third column is an example in human language to make it more understandable how the variations can be interpreted. For each dialog the objective and the result is the same despite of the variations.

1. *Original* – Control Test with exactly the same dialogs between r3 and r4 based upon ContractNet

r8: Cfp; *Can I have 200 boxes of Paracetamol per month?*

r7: Propose; *For that amount per month I offer you €20,- per box.*

r8: Accept-proposal; *That's a deal than!*

r7: inform-done; *Here is your receipt*

2. *Iterated* – Test with iterated FIPA-elements, using a conversation build upon ContractNet IP with iterated elements, therefore we use the Iterated-ContractNet IP which we call a dialect of ContractNet.

r8: Cfp; *Can I have 200 boxes of Paracetamol per month?*

r7: Propose; *For that amount per month I offer you €20,- per box.*

r8: Reject-proposal: *No, you can do better that that I know!*

r8: Cfp *Let me ask you again: Can I have 200 boxes of Paracetamol per month?*

r7: Propose; *For that amount per month I offer you €15,- per box*

r8: Accept-proposal; *That's a deal than!*

r7: Inform-done; *Here is your receipt*

3. *Nested* – Test of dialect with nested protocols, using a conversation build with ContractNet IP with a nested Request IP.

r8: Cfp; *Can I have 200 boxes of Paracetamol per month?*

r7: Propose; *For that amount per month I offer you €20,- per box.*

r8: Accept-proposal; *That's a deal than!*

r7: Request; *Sent me money to bank account 666.*

r8: Agree; *Ok!*

r7: inform-done; *Here is your receipt.*

4. *Noisy, replace* – Test of dialect which uses informative "request" in stead of "propose", based upon the dialog 1, between roles r3 and r4 in ContractNet.

r8: Cfp; *Can I have 200 boxes of Paracetamol per month?*

r7: Request;	*Sent me money to bank account 666.*
r8: Accept-proposal;	*That's a deal than!*
r7: inform-done;	*Here is your receipt*

5. *Noisy, add* – Test of dialect with the addition of the "S-Refuse" informative, based upon the conversation of role D in Contract Net.

r8: Cfp;	*Can I have 200 boxes of Paracetamol per month?*
r7: Propose;	*For that amount per month I offer you €20,- per box.*
r8: Refuse;	*I will not allow it!*
r8: Accept-proposal;	*On the other hand, ok, that's a deal than!*
r7: inform-done;	*Here is your receipt.*

6. *Noisy, remove* – Test of dialect, or incomplete overheard conversation, with the removal of the "call-for-proposal" informative, based upon the conversation of role D in Contract Net.

r7: Propose;	*For that amount per month I offer you €20,- per box.*
r8: Accept-proposal;	*That's a deal than!*
r7: inform-done;	*Here is your receipt.*

3.4 Recognition by deductive reasoning

3.4.1 Specification

Given the specification of symbols in section 3.2 we can setup the rule for strict recognition.

The rule of the deductive reasoning method says:

> An agent α from which δ^O has been overheard has role r if and only if $\delta^O \in \Delta_r$

For example: The role of an analyst r_{analyst} uses the query and contract net protocol from the participant perspective: π^P_{analyst} = {query, contractnet} and the request protocol from the initiator perspective: $\pi^{I}_{analyst}$ = {request}. All dialogs from these perspectives are contained in $\Delta_{analyst}$. When the observed dialog δ^{O} matches one of these dialogs in $\Delta_{analyst}$ it returns the value true, which means the dialog of α has been recognized to be typically used by an agent who enacts the role of an analyst. (the analyst role is the same as role id r3; the manager role is the same as role id r4)

3.4.2 Implementation; deductive reasoning with strict XML validation

We have implemented the deductive reasoning process by using XML. The process of validating XML documents is analog to deductive reasoning. XML is easy to create documents that resemble conversational structures like in chat sessions. Also XML is an attractive validation method is because it is widely used and most suitable for validating XML documents. The XML parser of delivers the opportunity to validate the structure of a document to a Document Type Definition (DTD) with ease.

Learning

In this phase we create characteristic information about the role r3.

- We consider the DTD as a representation of a protocol; in our case the protocol r3 and r4 use is the ContractNet protocol.
- In the learning phase the observer constructs a DTD that validates dialogs originating from r3 (the analyst). This DTD we call the ContractNet DTD.
- This ContractNet DTD is able to validate all dialogs the analyst role is permitted to create.

➔ Therefore this ContractNet DTD can be represented as $\Delta_{analyst}$.

The DTD validates an XML document. The XML document can be considered as a dialog δ.

In this method we provided a DTD in Table 7 which resembles the ContractNet Interaction Protocol that role r3 and r4 are using to communicate with each other in an interaction scene. With this DTD we are going to validate the overheard dialogs between role δ and γ. Therefore we create XML documents for each example dialog and validate it to the given DTD.

Table 7: DTD structure for ContractNet Interaction Protocol in the perspective of r3. To indicate r3 as the observed agent the α -symbol is used.

```
  <!ELEMENT dialog (cfp, (refuse | (propose, (reject-proposal | (accept-
proposal, (failure | inform-done | inform-result))))))>
  <!ELEMENT cfp               (#PCDATA)>
  <!ELEMENT refuse            (#PCDATA)>
  <!ELEMENT propose           (#PCDATA)>
  <!ELEMENT reject-proposal   (#PCDATA)>
  <!ELEMENT accept-proposal   (#PCDATA)>
  <!ELEMENT failure           (#PCDATA)>
  <!ELEMENT inform-done       (#PCDATA)>
  <!ELEMENT inform-result     (#PCDATA)>

<?The Attributes below define whether the initiator or the participant is
receiving or sending the communicative act.?>

<?Elements where the participant is typically the receiver?>
<!ATTLIST cfp receiver (α) #REQUIRED>
<!ATTLIST reject-proposal receiver (α) #REQUIRED>
<!ATTLIST accept-proposal receiver (α) #REQUIRED>
<?Elements where the participant is typically the sender?>
<!ATTLIST refuse sender (α) #REQUIRED>
<!ATTLIST propose sender (α) #REQUIRED>
<!ATTLIST failure sender (α) #REQUIRED>
<!ATTLIST inform-done sender (α) #REQUIRED>
<!ATTLIST inform-result sender (α) #REQUIRED>

<?Elements were the initiator is typically the sender?>
<!ATTLIST cfp sender CDATA "0">
<!ATTLIST reject-proposal sender CDATA "0">
<!ATTLIST accept-proposal sender CDATA "0">
<?Elements were the initiator is typically the receiver?>
<!ATTLIST refuse receiver CDATA "0">
<!ATTLIST propose receiver CDATA "0">
<!ATTLIST failure receiver CDATA "0">
<!ATTLIST inform-done receiver CDATA "0">
<!ATTLIST inform-result receiver CDATA "0">
```

More Interaction Protocols and their explanation can be found in Appendix section 8.3.

Testing

In the testing phase we are interested in the dialogs that are validated by the contractnet DTD. Each of the overheard dialogs are processed through the strict recognition rule stated above. The match is being made by the XML validator from XMLspy™.

The dialogs following below are the test conversations from section 3.3. These dialogs represent the conversation between r7 and r8 and are transformed into an XML format.

In this test the role r7 is observed by ω and the α-symbol is used to identify the target agent to follow in the conversation. (r7 is represented by α)

The XML documents are passed through the validator that validates the documents to the ContractNet DTD from Table 7.

Below the XML documents of the overheard dialogs are presented. The dialogs are analog to the test conversations presented in section 3.3. To keep the presentation simple, these XML documents are not made according to the FIPA XML specifications. In our specification each dialog consists of a root element <dialog>, followed by children. These children are elements named after a communicative act and contain sender and receiver attributes. This is also the format of the DTD.

Table 8: Dialog 1; Original conversation according to ContractNet IP

```
<dialog>
 <cfp sender="r8" receiver="α"/>
 <propose sender="α" receiver="r8"/>
 <accept-proposal sender="r8" receiver="α"/>
 <inform-done sender="α" receiver="r8"/>
</dialog>
```

Table 9: Dialog 2; Itterated, according to Itterated ContractNet IP

```
<dialog>
      <cfp sender="r8" receiver="α"/>
      <propose sender="α" receiver="r8"/>
      <reject-proposal sender="r8" receiver="α"/>
      <cfp sender="r8" receiver="α"/>
      <propose sender="α" receiver="r8"/>
      <accept-proposal sender="r8" receiver="α"/>
      <inform-done sender="α" receiver="r8"/>
</dialog>
```

Table 10: Dialog 3; Nested

```
<dialog>
      <cfp sender="r8" receiver="α"/>
```

```
      <propose sender="α" receiver="r8"/>
      <accept-proposal sender="r8" receiver="α"/>
      <request sender="α" receiver="r8"/>
      <agree sender="r8" receiver="α"/>
      <inform-done sender="α" receiver="r8"/>
</dialog>
```

Table 11: Dialog 4; Noise Replace

```
<dialog>
      <cfp sender="r8" receiver="α"/>
      <request sender="α" receiver="r8"/>
      <accept-proposal sender="r8" receiver="α"/>
      <inform-done sender="α" receiver="r8"/>
</dialog>
```

Table 12: Dialog 5; Noise Add

```
<dialog>
      <cfp sender="r8" receiver="α"/>
      <propose sender="α" receiver="r8"/>
      <refuse sender="r8" receiver="α"/>
      <accept-proposal sender="r8" receiver="α"/>
      <inform-done sender="α" receiver="r8"/>
</dialog>
```

Table 13: Dialog 6; Noise Remove

```
<dialog>
      <propose sender="α" receiver="r8"/>
      <accept-proposal sender="r8" receiver="α"/>
      <inform-done sender="α" receiver="r8"/>
</dialog>
```

3.4.3 Outcome

When passed through the validator only the first dialog passed. Passing the validator means that this dialog fits in the set of dialogs created from the source role.

The overheard dialogs can be either accepted or rejected that qualifies the agent to be a colleague of manager D:

- δ_1^O: Accepted → Validation Completed.
- δ_2^O: Rejected → Validated correctly until the second <cfp> element.
- δ_3^O: Rejected → Validated correctly until the <request> element; <failure>or<inform-done>or<inform-result> expected.
- δ_4^O: Rejected → Validated correctly until the <request> element; <refuse> or <propose> expected.
- δ_5^O: Rejected → Validated correctly until the <refuse> element; <reject-proposal> or <accept-proposal > expected.
- δ_6^O: Rejected → <cfp> element expected.

This demonstrates that a strict method such as deductive reasoning is not able to detect a dialect of the role. This could be expected because strictness is the nature of deductive reasoning.

3.5 Recognition with inductive reasoning

Here, we abandon the idea that communicative acts have to be in a specific order and rely on the probability a message *M* belongs to a role. A collection of *M* in a dialog δ will predict the role it belongs to.

3.5.1 Specification

For the rule of the loose recognition we use the equation of the Naive Bayesian Classifier:

An agent α from which δ^O has been overheard, the output is *r* from V_{NB} (see Equation 2) where:

- V is given by R (all learned roles)
- v_j iterates over every learned role $r_j \in$ R
- x_i are the messages M_i in δ^O

Learning

All the probabilities for P(r_3), P(r_4), P(M_i | r_3) and P(M_i | r_4) are calculated in advance in the learning phase where ω has been trained to distinguish role r3 from r4. This can be done by differentiating the messages from each perspective. The messages are tuples containing a communicative act and a send or received binary. For example in the learning phase the message accept-proposal appears to be received over and over again by the agent who enacts the role r3, so the P(<*accept-proposal, received*> | r_3) is high.

An explanation about how the probabilities are calculated for NB role recognition can be found below:

For P(M_1 | r_3)
In words: The chance that message M_1 can be uttered by role r_3.

Calculated as: $\#\{\delta \in \Delta r_3 \text{ where } M_1 \text{ occurs in } \Delta r_3\} / \#\{\delta \in \Delta r_3\}$
In words: The number of dialogs, that role r_3 can utter, which contains this M_1 message; divided by the total number of dialogs that role r_3 can utter.

For P(r_3)
In words: The chance that a role will be classified as r_3

Calculated as: $\#\{\delta \in \Delta r_3\} / \#\{\delta \in \Delta R\}$
In words: The number of all dialogs uttered by r_3 divided by the number of all dialogs from all roles (r_3 and r_4).

Testing
To output a role, for each overheard dialog the probability for each learned role is calculated. The messages used in this dialog represent a probability for each role. The collection of probabilities is picked from the table the agent created earlier in the learning phase and used to calculate the probability for a role of each overheard dialog.

For both roles the equation can be rewritten to:

$$P(r_3 \mid \delta^O) = P(r_3) \cdot P(M_1 \mid r_3) \cdot P(M_{\dots} \mid r_3) \cdot P(M_n \mid r_3)$$

And

$$P(r_4 \mid \delta^O) = P(r_4) \cdot P(M_1 \mid r_4) \cdot P(M_{\dots} \mid r_4) \cdot P(M_n \mid r_4)$$

For P(r_3 | δ^O) we are calculating the probability that the overheard dialog is similar to r_3.

For P(r_4 | δ^O) we are calculating the probability that the overheard dialog is similar to r_4.

The Naive Bayesian Classifier makes a distinction between two phases, the learning phase where the table of probabilities is created. The second phase is the testing phase where probabilities are selected based on the messages that appear in the overheard dialogs. These probabilities are used to calculate the probability of the role using that dialog. The output for an overheard message is the role (r_3 or r_4).with the highest probability.

3.5.2 Implementation

The calculation of the above probability tables in the learning and testing phase requires a lot of work. Therefore we have implemented the inductive reasoning using the NeuroSolutions™ Inc. neural network. [NeuroSolutions]

We made a data set that can be used by the neural network. This dataset contains both the training data and the test data. The task of the network is to train the characteristics of role r3 (the analyst) and non-r3 (the manager; r4). This trained network has been matched with the test data, which results in characteristics that are predicted through the network to be either r3 or non-r3. This is being matched with the original value and an error calculation is being made of the percentage correct answers, which shows the reliability of the predicted roles.

Learning

The train data are the dialogs permitted by ContractNet IP from both roles Δ_{r3} and $\Delta_{non\text{-}r3}$.

For the neural network we created a syntax to represent the tuple of a message. The transmission binary is followed by an underscore followed by a communicative act. The transmission binary can be either send, which is represented by an “S” or received which is represented by an “R”. The communicative acts can be one of the FIPA communicative act library. For example R_request.

For Δ each dialog is noted on a separate line and each message within a dialog is separated with a comma.

The dialogs used for training will be:

$\Delta_{role\ C}$:

$\delta 1$: R_cfp, S_refuse.

$\delta 2$: R_cfp, S_propose, R_reject-proposal.

$\delta 3$: R_cfp, S_propose, R_accept-proposal, S_failure.

$\delta 4$: R_cfp, S_propose, R_accept-proposal, S_inform-done.

$\delta 5$: R_cfp, S_propose, R_accept-proposal, S_inform-result.

$\Delta_{\text{non-role C}}$:

$\delta 1$: S_cfp, R_refuse.

$\delta 2$: S_cfp, R_propose, S_reject-proposal.

$\delta 3$: S_cfp, R_propose, S_accept-proposal, R_failure.

$\delta 4$: S_cfp, R_propose, S_accept-proposal, R_inform-done.

$\delta 5$: S_cfp, R_propose, S_accept-proposal, R_inform-result.

Testing

The test data is created in a similar way where we use the overheard dialogs 1 to 6 for input.

For example the overheard dialog 3 in the perspective of agent r7 is written $\delta_3{}^{O}{}_{r7}$ and looks like this:

R_cfp, S_proposal, R_accept-proposal, S_request, R_agree, S_inform-done.

Providing this input the network is able to classify this overheard conversation as for example an agent acting as role r3.

To indicate the reliability of this classification, each perspective of the overheard dialog was labeled. The label indicates that a perspective of an overheard dialog is either typically uttered by role r4 or r3, which is desired by “us”, the user. In NeuroSolutions™ this reliability is measured in the mean squared error (MSE), which is the distance between the classified output and the desired output. The desired output is the probability generated by the test data from both Δ_{r3} and $\Delta_{\text{non-r3}}$. The classified output V_{NB} is either r3 or r4 from the argmax of $P(r_3 \mid \delta^{O})$ and $P(r_4 \mid \delta^{O})$.

In the overheard dialogs we consider that agent r8 performs similar behavior as role r4 and agent r7 in the pharmaceutical company performs similar behavior as role r3, therefore we labeled all r8 conversations as role r4, and all r7 conversations as role r3. Please note that is for testing and control purpose, and is not involved in the learning phase.

More details about the training and testing data can be found in the appendix section 1.1.

3.5.3 Outcome

The overheard dialogs can be either accepted or rejected to qualify α to be similar to r3. The lower the MSE the more reliable the classified outcome is. Every bullet is an overheard dialog. Within the overheard dialog the separate perspectives of the agents on the dialog are classified by NeuroSolutions™. The first sub-bullet is the dialog in the perspective of agent r7, the second sub-bullet is the dialog perspective on the side of agent r8.

- δ_1^O: Accepted →
  - $\delta_1^O{}_{r7}$ has been labeled as "role r3"; the classifier has recognized this dialog to be similar to role r3, with a mean square error of $1{,}14 \cdot 10^{-05}$
  - $\delta_1^O{}_{r8}$ has been labeled as "role r8"; the classifier has recognized the dialog perspective to be similar to role r8, with a mean square error of $3{,}74 \cdot 10^{-06}$

- δ_2^O: Accepted →
  - $\delta_2^O{}_{r7}$ has been labeled as "role r3"; the classifier has recognized the dialog perspective to be similar to role r3, with a mean square error of 0,13
  - $\delta_2^O{}_{r8}$ has been labeled as "role r8"; the classifier has recognized the dialog perspective to be similar to role r8, with a mean square error of 0,26

- δ_3^O: Accepted →
  - $\delta_3^O{}_{r7}$ has been labeled as "role r3"; the classifier has recognized the dialog perspective to be similar to role r3, with a mean square error of $1{,}36 \cdot 10^{-03}$
  - $\delta_3^O{}_{r8}$ has been labeled as "role r8"; the classifier has recognized the dialog

perspective to be similar to role r8, with a mean square error of $2{,}59\cdot10^{-03}$

- δ_4^O: Accepted →
    - $\delta_4^O{}_{r7}$ has been labeled as “role r3”; the classifier has recognized the dialog perspective to be similar to role r3, with a mean square error of $5{,}26\cdot10^{-03}$
    - $\delta_4^O{}_{r8}$ has been labeled as “role r8”; the classifier has recognized the dialog perspective to be similar to role r8, with a mean square error of $7{,}19\cdot10^{-03}$

- δ_5^O: Accepted →
    - $\delta_5^O{}_{r7}$ has been labeled as “role r3”; the classifier has recognized the dialog perspective to be similar to role r3, with a mean square error of $2{,}69\cdot10^{-04}$
    - $\delta_5^O{}_{r8}$ has been labeled as “role r8”; the classifier has recognized the dialog perspective to be similar to role r8, with a mean square error of $9{,}05\cdot10^{-05}$

- δ_6^O: Accepted →
    - $\delta_6^O{}_{r7}$ has been labeled as “role r3”; the classifier has recognized the dialog perspective to be similar to role r3, with a mean square error of $1{,}01\cdot10^{-01}$
    - $\delta_6^O{}_{r8}$ has been labeled as “role r8”; the classifier has recognized the dialog perspective to be similar to role r8, with a mean square error of $4{,}33\cdot10^{-02}$

More detailed results about the six overheard dialogs can be found in the appendix section 8.6. In Illustration 20 we put the MSE’s from the overheard dialogs in a diagram below.

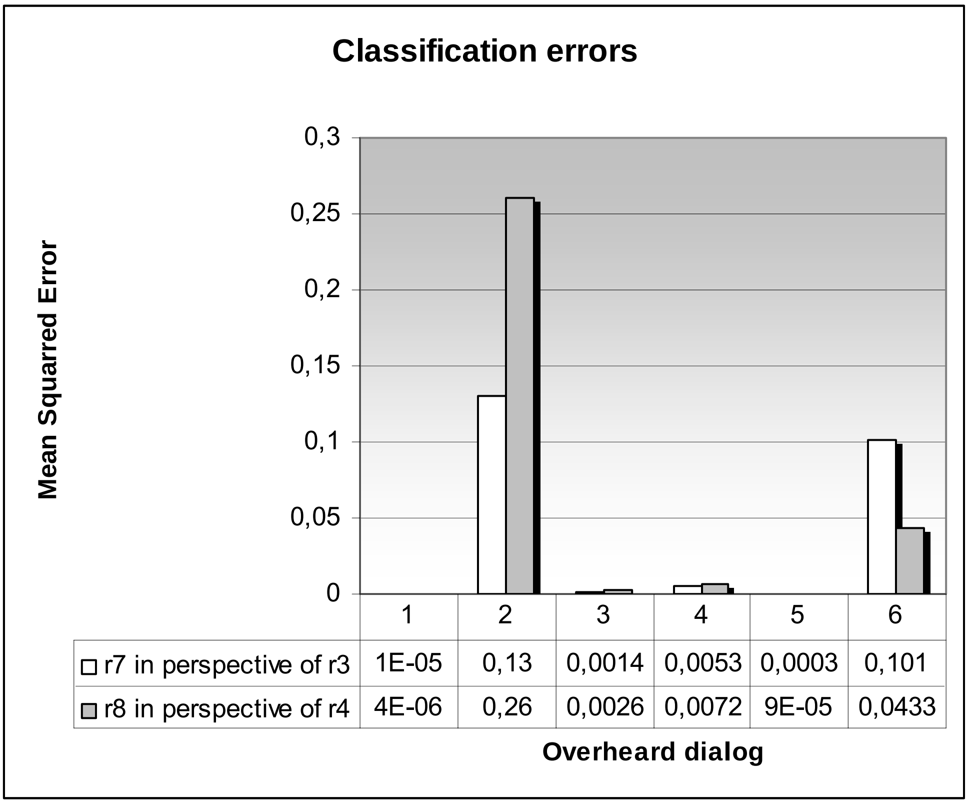


Illustration 20: Mean Squared Error, between the predicted and actual class in classifying r3 and r4

Illustration 20 shows that the overheard dialog 2 (iterated) and the overheard dialog 6 (noisy, remove) have a higher MSE value than the rest. This indicates that the classifications of the other overheard dialogs are more reliable than of 2 and 6.

3.6 Summary

In strict recognition the XML validation is able to find a matching conversational partner. It will not find similarities in the sequenced dialog and output a probability for a class.

In loose recognition the Naive Bayesian classifier succeeds in classifying the overheard dialogs.

4 Summary

The essence of a role is defined by several internal properties; name, objectives, norms, rights and type (research task 1). The expressions of these properties are externalized through the use of communication (research task 3). In multi-agent systems several agents work together, to define their function and workspace, roles are introduces (research task 2). In order to cooperate role-enacting agents have to communicate. In multi-agent systems FIPA-Agent Communication Language is the standard for verbal communication. The FIPA-ACL provides a set of Communicative Acts that Software agents can use to create sentences (research task 3). The set of Communicative Acts agents use in a communication dialog represent the essence of the role they enact (research task 1&3).

This set of Communicative Acts is quantifiable so that they can be used in computational recognition methods. Recognition requires two phases; a learning phase and a testing phase. In the learning phase observed Communicative Acts are stored into memory. In the testing phase observed Communicative Acts are compared to the one in memory. This testing is typically a reasoning process which can be split into two reasoning methods; deductive inference and inductive inference. Inference refers to the process of creating new knowledge from what already is known. Deductive inference leads to a result that is mathematically exact. Inductive inference leads to a result that is a probability modeled with Bayes' theorem (research task 4).

The implementation used for deductive reasoning is XML document validation, where in the learning phase a DTD is created. In the testing phase an observed conversation is put into an XML document which is validated according the DTD. The implementation used for inductive reasoning is a Naive Bayesian Classifier, where in the learning phase a probability table is created where independent values for communicative acts represent a role. In the testing phase the probability for every role is calculated from the observed communicative acts (research task 5&6).

5 Conclusion

How can a software agent recognize roles in a multi-agent system? The “How” in this question can be interpreted in two ways: The first way describes the process of recognition and the second way describes the practical implementation of this process. The first part specifies that a deductive or inductive reasoning process is required to identify a role. When the recognition process has to be done based upon verbal communication between role-enacting agents, communicative acts are recommended to use as an input value. In the second part implementation methods of these processes, involving communicative acts, show how role identification can be implemented for a multi-agent system. The agent who wants to recognize roles acts as an observer. This observer can learn and recognize roles using the Naive Bayesian classification method as the recommended reasoning process.

(Semi-)Autonomously recognizing roles can be used in multi-agent systems where social relations between two role-enacting agents have not been constructed earlier. Picturing a world where distributed agents share knowledge, as described in Bonifacio (2002). In this world agents contact each other based on the matching roles they enact. The recognition of roles is useful to discover knowledge in unforeseen areas were new social relations have not been matched earlier. This provides the agent the ability to go beyond the boundaries of an agent organization and expand the field of his knowledge contributions to a larger area.

6 Discussion & Future research

In this section we will discuss the results and implications of the reasoning methods along with the future work and future possibilities on role recognition.

Recognition with Content or Communicative acts: With the current recognition method, based on communicative acts, a sales man can be distinguished from a buyer. However this method does not distinguishes between a sales man who sells cars and a sales man who sells bicycles. This subject to recognize domain specific information can be used for future research, using a naive Bayesian classifier is recommended to classify texts, just like spam can be filtered from e-mail.

Differences in reasoning methods: There is an explanation for the discrepancy between the deductive and inductive reasoning method. Deductive reasoning requires an exact match between training data and test data. The variations made in the test conversations did not match exactly the training data and therefore the conversations seem to appear not to be socially compatible. However this capability of exact matching is useful for detecting agents who use similar protocols.
Inductive reasoning is not a method to indicate 100% accuracy for detecting agents who use a similar protocol. However it operates with a certain degree of freedom, this method is useful to detect agents who perform social similarities.

It has not been tested what kind of dialogs the Naive Bayesian classifier (NBC) has to observe to show a result where a dialog cannot be classified: The result of all test conversations with the NBC indicated that all roles are correctly classified. In our test conversation set we did not include dialogs that are originating from a completely different protocol. Also, we did not include observed dialogs originating from a different protocol in our testing set. In future research we can extend the training and testing dataset with more dialogs created from other interaction protocols.

NBC and unstructured vs. structured messages as independent variables: In the current NBC test we calculated the probability that a message (M_i) belongs to a role independently of the location of that message in the dialog. This means that the current method recognizes a dialog with randomly placed messages. Compared to the XML validation method this presumption is somewhat too flexible. In future research we can provide some degree of structure by combining two messages as to be considered one independent variable. $<M_i,M_j>$. For a dialog containing three messages (M_1, M_2 and M_3), combinations of pairs are made which resolves in six independent variables. These six variables are used for training and testing with the NBC.

Erosion of information in NBC: It appears in a check, that when communicative acts (like request and agree) are not learned in the training phase, the NBC will strip them from the dialog it has observed. The only variables it calculated its probability with are the ones that appear in the data table. The influence of this behavior on the results is unknown. We could say that if some thing that has not been conceptualized in the world of phenomena, it simply does not exist.

Universal Role Ontology: In future research a universal role identifier has to be taken into account. This special identifier reveals the essence of a role-enacting agent by an identifier in stead of overhearing conversations. This universal role identifier refers to the location of a role group in a universal role ontology. For an agent to find a social compatible role, it has to look for other agents with a closely related universal role identifier.

Theory and practice: this thesis withholds only conclusions based on a theoretical framework. The results are not tested with a real working multi-agent system. It provides only a lightweight roadmap how to recognize roles when an application is required to do so.

7 References

[Altova] XML validator; 21 April 2006; http://www.altova.com

[Anderson, 2000] Anderson, J.R., *Cognitive psychology and its implications*, 2000, Worth, London, New York, ISBN 0-7167-3678-0

[arafa, 2003] Arafa, Y., Mamdani, A., Scripting embodied agents behavior with CML: character mark-up language, 2003, *Proceedings of the 8th international conference on intelligent user interfaces*, pages 313-316, ACM Press, New York, ISBN: 1-58113-586-6, bookmark link: http://doi.acm.org/10.1145/604045.604109

[Aston, 1991] Aston, E. and Savona, G., *Theatre as sign-system*, 1991, Routledge and Kegan Paul, London, ISBN 0-415-04931-8 / 0-415-04932-6

[Biddle, 1979] Biddle, B.J., *Role theory: expectations, identities and behaviors*, 1979, Academic Press, New York, ISBN 0-12-095950-X

[Bonifacio, 2002] Bonifacio, M., Bouquet, P., Traverso, P., Distributed Knowledge Management: a Systemic Approach, In Gianfranco Minati, Eliano Pessa (Eds.). *Emergence in Complex, Cognitive, Social and Biological Systems*, Kluwer Academic/Plenum Publishers, New York, 2002, ISBN 0306473585

[Bonifacio, 2003] Bonifacio, M., Bouquet, P., Mameli, G. and Nori, M., Peer-Mediated Distributed Knowledge Management, *Agent-Mediated Knowledge Management: International Symposium AMKM 2003*, Stanford, CA, USA, March 24-26, Lecture Notes in Computer Science, Volume 2926 / 2003, Springer Berlin / Heidelberg, ISBN 3-540-20868-2, ISSN 0302-9743

[Borgelt] Borgelt, C.; 03-02-2006; http://fuzzy.cs.uni-magdeburg.de/~borgelt/doc/hopfield/

[Clark, 1996] Clark, H.H., *Using Language*, 1996, Cambridge University Press, Cambridge, ISBN: 0-521-56745-9

[Darlington, 2000] Darlington, K., *The essence of Expert Systems*, 2000, Perlson Education Ltd, Harlow, ISBN: 0-13-022774-9

[Davies, 1998] Davies, N.J., Stewart, R.S. and Weeks, R., Knowledge Sharing Agents Over the World Wide Web, *BT Technology Journal, Volume 16, Number 3*, Publisher: Springer Netherlands, 1998, ISSN 1573-1995

[Dignum, 2004] Dignum, V., *A Model for Organizational Interaction*, Phd. Thesis, Dutch Research School for Information and Knowledge Systems, Utrecht, 2004, ISBN 90-3930-3568-0

[Duba et al, 2000] Duba, R.O., Hart, P.E., Stork, D.G., *Pattern classification*, 2nd edition, John Wiley & Sons, 2001, 605 Third Avenue, New York, ISBN: 0471056693

[Dunham, 2003] Margaret H. Dunham, *Data mining – introductory and advanced topics*, 2003, Prentice Hall, Upper Saddle River, NJ 07458, ISBN: 0-13-088892-3

[Elam, 1980] Elam, K., *The semiotics of theatre and drama*, 1980, Methuen, London, ISBN 0-416-72060-9 / 0-416-72050-1

[Elst, 2003] Elst, van L., Dignum,V., Abecker A., Towards Agent-Mediated Knowledge Management, *Agent-Mediated Knowledge Management: International Symposium AMKM 2003*, Stanford, CA, USA, March 24-26, Lecture Notes in Computer Science, Volume 2926 / 2003, Springer Berlin / Heidelberg, ISBN 3-540-20868-2, ISSN 0302-9743

[FIPA] Foundation of Physical Agents; 03-02-2006; http://www.fipa.org

[Grenander, 1993] Ulf Grenander, *General Pattern Theory*, 1993, Clarendon Press, Oxford, ISBN 0-19-853671-2

[HML] Human Mark-up Language; http://www.humanmarkup.org/

[HMM] Hidden Markov Model; 16 jan. 2006; http://en.wikipedia.org/wiki/Hidden_Markov_model

[Hugin] Hugin Case visualization, 23-12-2005, http://www.hugin.com/

[Levinson, 1983] Levinson, S. C., *Pragmatics*, 1983, Cambridge University Press, Cambridge, ISBN 0-521-29414-2.

[Mathieu, 2002] Mathieu, P. and Routier, J.C. and Secq, Y., Dynamic organization of multi-agent systems, 2002. *Proceedings of the first international joint conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, http://eprints.agentlink.org/4649/

[McTear, 2002] McTear, M.F., Spoken dialogue technology: enabling the conversational user interface, 2002, *ACM Computing Surveys, volume 34, Issue 1*, pages 90-196, ACM Press, New York, NY, ISSN: 0360-0300, bookmark link:

http://doi.acm.org/10.1145/505282.505285

[Mitchell, 1997] Tom Mitchell, *Machine Learning*, International Edition, 1997, McGraw-Hill, Singapore, ISBN: 0-07-115467-1

[NeuroSolutions] NeuroSolutions™ Inc. neural network; 21 April 2006; http://www.neurosolutions.com

[Odell, 2002] Odell, J. and Van Dyke-Parunak, H., Representing Social Structures in UML, *Agent-Oriented Software Engineering II, Second International Workshop, AOSE 2001*, Montreal, Canada, May 29, 2001, Lecture Notes in Computer Science , Vol. 2222, Springer, 2002, ISBN 3540432825

[Rosengren, 2000] Rosengren, K.E., *Communication - an introduction*, 2000, Sage Publications Ltd, London, ISBN 0-8039-7836-7

[talkingheads] VHML example; 13-12-2005; http://www.talkingheads.computing.edu.au/documents/workshop/TalkingHeadsTechnologyWorkshop/

workshop/marriott/vhml_workshop.pdf

[VHML] Virtual Human Mark-up Language; http://www.vhml.org/

[wikipedia, dialect] Dialect; 15 feb. 2006; http://en.wikipedia.org/wiki/Dialects

[wikipedia, essence] Essence; 17 may; http://en.wikipedia.org/wiki/Essence

[wikipedia, organization] Organization, 15 feb. 2006; http://en.wiktionary.org/wiki/organization

[wikipedia, phenomenon] Phenomenon; 17 may; http://en.wikipedia.org/wiki/Phenomenon

[wiktionary, protocol] Protocol; 15 feb. 2006; http://en.wiktionary.org/wiki/Protocol

[Wooldridge, 2002] Wooldridge, M.J. *An introduction to multiagent systems*, 2001, John Wiley & Sons Ltd, Chichester, ISBN 0-471-49691-X

[Wooldridge, 2005] Wooldridge, M.J., *Research Interests*; 07-12-2005; http://www.csc.liv.ac.uk/~mjw/research/

8 Appendix

8.1 FIPA ACL message parameters

Table 14: FIPA ACL Message Parameters

Parameter	Category of Parameters
performative	Type of communicative acts
sender	Participant in communication
receiver	Participant in communication
reply-to	Participant in communication
content	Content of message
language	Description of Content
encoding	Description of Content
ontology	Description of Content
protocol	Control of conversation
conversation-id	Control of conversation
reply-with	Control of conversation
in-reply-to	Control of conversation
reply-by	Control of conversation

From http://www.fipa.org/repository/aclspecs.html

8.2 FIPA Communicative Acts

The language used in FIPA is called Agent Communication Language (ACL), which consists of a syntax and semantics. The syntax is the set of communicational elements, or Communication Acts, or Speech Acts. In FIPA these are defined in the Communicative Acts Library (CAL)

The Communicative Acts are in this concern the atomic elements upon which feature analysis can take place. The following list will provide all valid performatives:

accept-proposal, agree, cancel, cfp, confirm, disconfirm, failure, inform, not-understood, propose, query-if, query-ref, refuse, reject-proposal, request, request-when, request-whenever, subscribe, inform-if, inform-ref, proxy, propagate.

From http://www.fipa.org/specs/fipa00037/index.html

An example for the definition of the Call For Proposal performative is given below, which is an exact copy of the standard FIPA CAL document found on www.fipa.org

Call For Proposal

Summary	The action of calling for proposals to perform a given action.
Message Content	A tuple containing an action expression denoting the action to be done, and a referential expression defining a single-parameter proposition which gives the preconditions on the action.
Description	cfp is a general-purpose action to initiate a negotiation process by making a call for proposals to perform the given action. The actual protocol under which the negotiation process is established is known either by prior agreement or is explicitly stated in the protocol parameter of the message. In normal usage, the agent responding to a cfp should answer with a proposition giving the value of the parameter in the original precondition expression (see the statement of rational effect for cfp). For example, the cfp might seek proposals for a journey from Frankfurt to Munich, with a condition that the mode of travel is by train. A compatible proposal in reply would be for the 10.45 express train. An incompatible proposal would be to travel by airplane. Note that cfp can also be used to simply check the availability of an agent to perform some action. Also note that this formalization of cfp is restricted to the common case of proposals characterized by a single parameter (x) in the proposal expression. Other scenarios might involve multiple proposal parameters, demand curves, free-form responses, and so forth.
Formal Model	<i, cfp (j, <j, act>, Ref x φ(x))> ≡ <i, query-ref (j, Ref x (I_i Done (<j, act>, φ(x)) (I_j Done (<j, act>, φ(x))))> FP: ¬$Bref_i$(Ref x α(x)) ∧ ¬$Uref_i$(Ref x α(x)) ∧ ¬$B_i I_j$Done (<j, inform-ref (i, Ref x α(x))>) RE: Done (<j, inform (i, Ref x α(x) = r1)> | … | <j, inform (i, Ref x α(x) = r_k)>) Where: α(x) = IiDone (<j, act>, •(x)) IjDone (<j, act>, •(x)) Agent *i* asks agent j: "What is the '*x*' such that you will perform action '*act*' when 'φ(x)' holds?" Note: Ref x •(x) is one of the referential expressions: ιx •(x), any x •(x) or all x •(x). Note: The rational effect of this is not a proposal by the recipient. Rather, it is the value of the proposal parameter. See the example in the definition of the proposeact.
Example	Agent j asks i to submit its proposal to sell 50 boxes of plums. (cfp :sender (agent-identifier :name j) :receiver (set (agent-identifier :name i)) :content "((action (agent-identifier :name i) (sell plum 50)) (any ?x (and (= (price plum) ?x) (< ?x 10))))" :ontology fruit-market :language fipa-sl)

8.3 FIPA protocols

With the use of these performatives (communicative acts) one is able to make a conversation. If a conversation is conducted each time in the same way we can speak of a protocol. If this repetitive conversation is conducted by one role only then the protocol is characteristic for this role. We speak of a role specific protocol.

A protocol can be classified for different environments that the describe the dependencies between two roles, also called coordination types [Dignum, p. 64]. The table below shows the different coordination types for protocols to be written.

	Market	*Network*	*Hierarchy*
Type of society	Open	Trust	Closed
Agent 'values'	Self interest	Mutual interest / Collaboration	Dependency
Facilitation roles	Matchmaking Banking	Gate keeping Registry Matchmaking	Interface Control
Dependency relation	Bidding	Request	Delegation

Table 15: Coordination types [Dignum, p.64]

FIPA also made protocols that apply to a market environment. Such as :

Query Interaction protocol, *ContractNet* Interaction protocol, *Itterated ContractNet* protocol, *Brokering* Intraction protocol, *Recruiting* Interaction protocol, *Propose* Interaction Protocol, *Request* Interaction protocol, *Request-When* Interaction protocol, *Subscribe* Interaction protocol and *Cancel Meta* Interaction protocol.

8.4 Putting FIPA in XML

These protocols have an interaction structure that looks pretty much the same as the structure definitions in a DTD (Document Type Definition). Which means that recognizing a role specific protocol is like validating an XML document. By using a DTD we are able to show the structure of a protocol, and therefore the patterns that might occur in a conversation. (If the conversation is in XML FIPA ACL)

Below all available FIPA protocols are written in a pseudo DTD format just to have a collection as a starting point to the next sections. In the logical rule structure DTD: Grey illocutions represent “end of the line” illocutions. The subscripted letters mean Illocutions belong to a Role. Summary of signs: | means “or”, + means “one or more”, * means “zero or more”, a comma represents the sequence of steps.

Query Interaction Protocol

Initiator	*Participant*
Query-if | query-ref	refuse | agree
	failure | inform-t/f:inform | inform-result:inform

Logical Structure (DTD)
(query-if | query-ref)$_I$, (refuse | agree, (failure | inform-t/f | inform-result))$_P$

Contract Net Interaction Protocol

Initiator	*Participant*
Cfp	refuse | propose
reject-proposal | accept-proposal	failure | inform-done:inform |inform-result:inform

Logical Structure
cfp$_I$, (refuse$_P$ | (propose$_P$, (reject-proposal$_I$ | (accept-proposal$_I$, (failure | inform-done | inform-result)$_P$))))

Itterated Contract Net Interaction Protocol

Initiator	*Participant*
cfp-[n-1]:cfp	refuse | propose
reject-proposal-[n-1]:reject-proposal | cfp-[n]:cfp	failure | inform-done:inform |inform-result:inform
reject-proposal-[n]:reject-proposal | accept-proposal	

Logical Structure (DTD)
(cfp$_I$, (refuse | propose)$_P$)+, (reject-proposal$_I$ | (accept-proposal$_I$, (inform | failure)$_P$)?

Brokering Interaction Protocol

Initiator	*Broker*
Proxy	refuse | agree
	(failure-no-match) | (failure-proxy:failure | inform-done-proxy:inform)
	reply-message-sub-protocol | failure-brokering:failure

Logical structure (DTD)
proxy$_I$, (refuse | (agree, (failure-no-match | (failure-proxy | (inform-done-proxy, (reply-message-sub-protocol | failure-brokering))))))$_B$

Recruiting Interaction protocol

Initiator	*Recruiter*	*Designated Receiver*	*Target*

Initiator	***Recruiter***	***Designated Receiver***	***Target***
Proxy	refuse | agree	(only gets messages)	Reply-message-sub-protocol
	failure-no-match		
	failure-proxy | inform-done-proxy		

Logical Structure (DTD)
proxy$_I$, (refuse$_R$ | (agree$_R$, (failure-no-match$_R$ | (failure-proxy$_R$ | (inform-done-proxy$_R$, (reply-message-sub-protocol | failure-brokering)$_T$)))))

Propose Interaction Protocol

Initiator	***Participant***
Propose	reject-proposal | accept-proposal

Logical Structure (DTD)
propose$_I$, (reject-proposal | accept-proposal)$_P$

Request Interaction Protocol

Initiator	***Participant***
Request	refuse | agree
	failure | inform-done | inform-result

Logical Structure (DTD)
request$_I$, (refuse | agree, (failure | inform-done | inform-result))$_P$

Request-When Interaction Protocol

Initiator	***Participant***
request-when	refuse | agree
	failure | inform-done | inform-result

Logical Structure (DTD)
request-when$_I$, (refuse | agree, (failure | inform-done | inform-result))$_P$

Subscribe Interaction Protocol

Initiator	***Participant***
Subscribe	refuse | agree
	inform-result+ | failure

Logical Structure (DTD)
subscribe$_I$, (refuse | agree, (failure | inform-result+))$_P$

Cancel Meta Interaction Protocol

Can be used anywhere, any time in the conversation.

Initiator	***Participant***
Cancel	Inform-done | failure

Logical Structure (DTD)

$\texttt{cancel}_I$, $(\texttt{inform-done} \mid \texttt{failure})_P$

8.5 Naive Bayes Classifier INPUT

					Classes		FIPA-ACL elements																			
Data Type	Dialog track	Protocol Meta-Rol	Dialoog	Dialog-Type	is Manager	is Role C	S-Cfp	R-Cfp	S-Request	R-Request	S-Propose	R-Propose	S-Accept-proposal	R-Accept-proposal	S-Reject-proposal	R-Reject-proposal	S-Agree	R-Agree	S-Refuse	R-Refuse	S-Failure	R-Failure	S-Inform-done	R-Inform-done	S-Inform-result	R-Inform-result
TRAIN	1	Participant	1	ContractNet	***0***	***1***	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
TRAIN	2	Initiator	1	ContractNet	***1***	***0***	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
TRAIN	3	Participant	2	ContractNet	***0***	***1***	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
TRAIN	4	Initiator	2	ContractNet	***1***	***0***	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
TRAIN	5	Participant	3	ContractNet	***0***	***1***	0	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
TRAIN	6	Initiator	3	ContractNet	***1***	***0***	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
TRAIN	7	Participant	4	ContractNet	***0***	***1***	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
TRAIN	8	Initiator	4	ContractNet	***1***	***0***	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
TRAIN	9	Participant	5	ContractNet	***0***	***1***	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
TRAIN	10	Initiator	5	ContractNet	***1***	***0***	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
TEST 1	9	Participant	5	ContractNet	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
TEST 1	10	Initiator	5	ContractNet	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
TEST 2	11	Participant	2	ItteratedCN	0	1	0	2	0	0	2	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0
TEST 2	12	Initiator	2	ItteratedCN	1	0	2	0	0	0	0	2	1	0	1	0	0	0	0	0	0	0	0	1	0	0
TEST 3	13	Participant	3	Nested	0	1	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0
TEST 3	14	Initiator	3	Nested	1	0	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0
TEST 4	15	Participant	4	noisy, replace	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
TEST 4	16	Initiator	4	noisy, replace	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
TEST 5	17	Participant	5	noisy, add	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0
TEST 5	18	Initiator	5	noisy, add	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0
TEST 6	19	Participant	6	noisy, remove	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
TEST 6	20	Initiator	6	noisy, remove	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0

Legend for the table above

The White Fields is Training Data, and the grey fields are Test data.

The Training data is generated using the ContractNet IP where every possible dialog is written down in the perspective of role C as well as for role D.

For reference with the text in the thesis: Role C = r_3 *and Role D =* r_4*.*

The Test data is created using the Example dialogs, also in the two different perspectives.

The first five columns are not taken into the calculation; they only give extra information to the reader:

Data type: Discriminates the rows when they are used for training or testing.

Dialog-Track: increments the number if the conversation perspective is a unique one in the table.

Meta-Role: Indicates who starts the dialog.

Dialog: a pair of dialog perspectives that belong to each other.

Dialog-Type: describes the protocol or test-dialect in which the dialog has been created from.

8.6 Naive Bayes Classifier OUTPUT

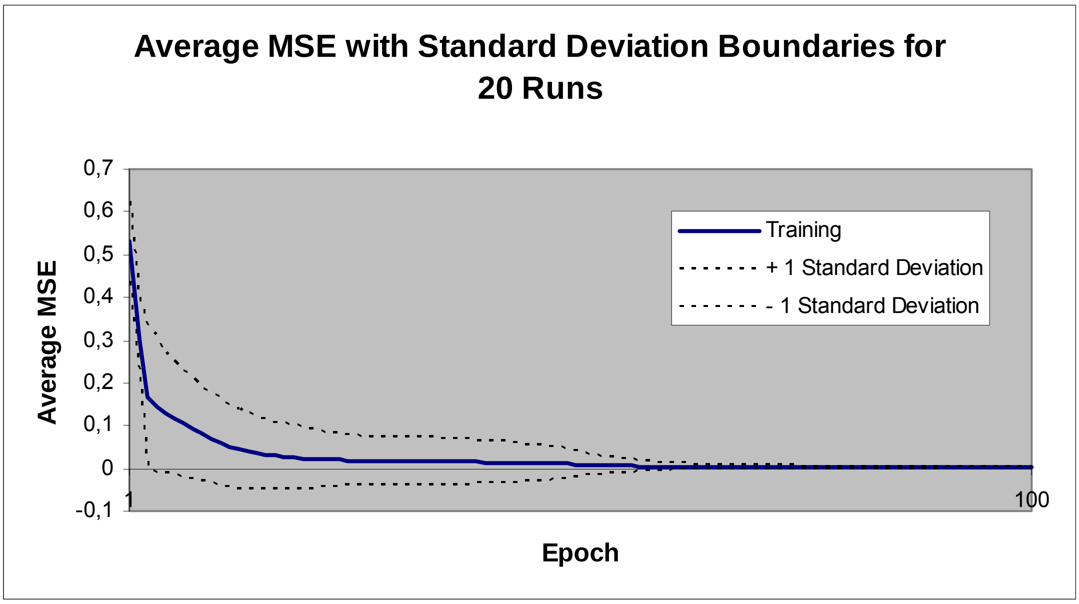
All data has been processed by NeuroSolutions for Excel version 5.0.

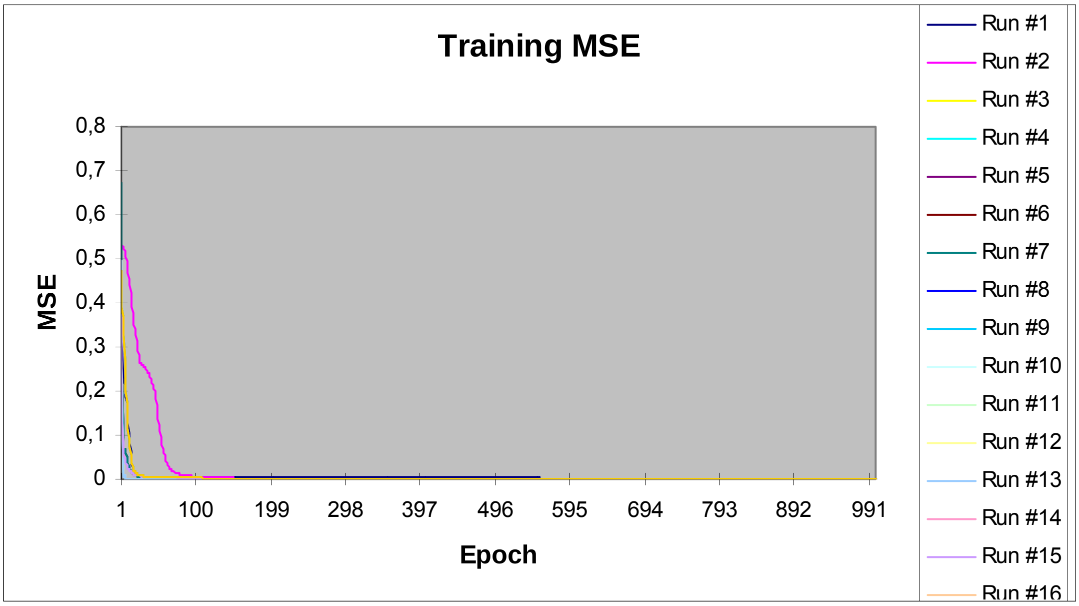
8.6.1 Training

I trained the network from the given information ContractNet IP conversations as shown in the above table.

All Runs	Training Minimum	Training Standard Deviation
Average of Minimum MSEs	0,000622989	0,000540437
Average of Final MSEs	0,000622989	0,000540437

Best Network	Training
Run #	14
Epoch #	1000
Minimum MSE	7,58361E-06
Final MSE	7,58361E-06

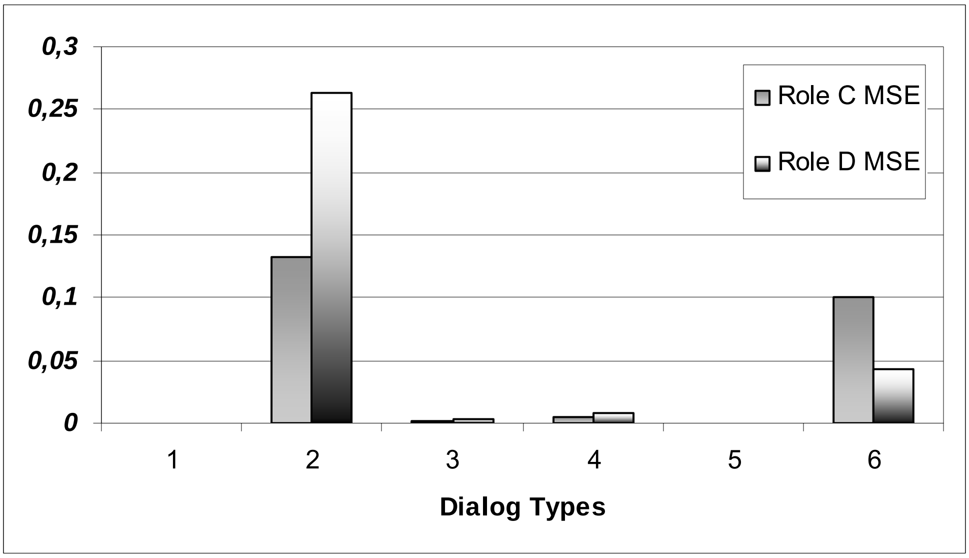




8.6.2 Testing

The certainty of the found role can be expressed by the Mean Squared Error. The lower the error the more certain the network is to classify a dialog as a role.

Dialog Type	Role C MSE	Role D MSE
1	1,1411E-05	3,7417E-06
2	0,13264437	0,26391605
3	0,00135938	0,00259117
4	0,00526343	0,00719065
5	0,00026881	9,0498E-05
6	0,10055263	0,04331053



Below, the results of every test 1 till 6 will be displayed.

Results Test 1

Output / Desired	*is Manager*	*is Role C*
is Manager	1	0
is Role C	0	1

Performance	***is Manager***	***is Role C***
MSE	3,74172E-06	1,14107E-05
NMSE	1,49669E-05	4,56429E-05
MAE	0,001934118	0,003575436
Min Abs Error	0,00034537	0,000954218
Max Abs Error	0,001595355	0,004262536
r	1	1
Percent Correct	100	100

Results Test 2

Output / Desired	*is Manager*	*is Role C*
is Manager	1	0
is Role C	0	1

Performance	***is Manager***	***is Role C***
MSE	0,263916053	0,132644367
NMSE	1,055664211	0,530577468
MAE	0,289858669	0,171127013
Min Abs Error	0,012926577	0,026048695
Max Abs Error	0,051398515	0,026346661
r	1	1
Percent Correct	100	100

Results Test 3

Output / Desired	*is Manager*	*is Role C*
is Manager	1	0
is Role C	0	1

Performance	***is Manager***	***is Role C***
MSE	0,00259117	0,001359376
NMSE	0,01036468	0,005437506
MAE	0,051198639	0,031577092
Min Abs Error	0,009124401	0,00068082
Max Abs Error	0,058129114	0,011274725
r	1	1
Percent Correct	100	100

Results Test 4

Output / Desired	*is Manager*	*is Role C*
is Manager	1	0
is Role C	0	1

Performance	***is Manager***	***is Role C***
MSE	0,007190646	0,005263433
NMSE	0,028762584	0,021053732
MAE	0,095570994	0,083467849
Min Abs Error	0,043882378	0,015810771
Max Abs Error	0,098098553	0,055553933
r	1	1
Percent Correct	100	100

Results Test 5

Output / Desired	*is Manager*	*is Role C*
is Manager	1	0
is Role C	0	1

Performance	***is Manager***	***is Role C***
MSE	9,04985E-05	0,000268812
NMSE	0,000361994	0,001075247
MAE	0,010992606	0,017101139
Min Abs Error	0,004371874	0,002176677
Max Abs Error	0,006992796	0,021032995
r	1	1
Percent Correct	100	100

Results Test 6

Output / Desired	*is Manager*	*is Role C*
is Manager	1	0
is Role C	0	1

Performance	***is Manager***	***is Role C***
MSE	0,043310534	0,100552631
NMSE	0,173242136	0,402210523
MAE	0,23430579	0,327351137
Min Abs Error	0,162301699	0,023881139
Max Abs Error	0,234662345	0,396515345
r	1	1
Percent Correct	100	100