DISSERTATION

Generation of Inner and Outer Speech by Means of Situational Context

Submitted at the Faculty of Electrical Engineering, Vienna University of Technology in partial fulfillment of the requirements for the degree of Doctor of Technical Sciences

under supervision of

Prof. Dr. Dietmar Dietrich
Institute number: 384
Institute of Computer Technology Vienna University of Technology

and

Ph.D. Oscar Corcho
Departamento de Inteligencia Artificial
Facultad de Informática, Universidad Politécnica de Madrid

by

Mag. Dipl.-Ing. Isabella Hinterleitner, M.Sc. Matr.Nr. e0006380
Zierenbergstrasse 73, D-31137 Hildesheim

June 2014 (Version 15) ______________________________

Abstract

The work solves the question of establishing semantic and situational context in project ARS (Artificial Recognition System). Context is needed in ARS, if two agents have to solve a question in a certain problem domain. It is defined through a domain specific ontology and a general ontology. Talking about a distinctive problem requires agents to know the problem domain and to have specific knowledge about it. Solving a general problem requires less domain specific knowledge, but more abstract knowledge. For that reason, two different ways of knowledge bases exist.

In order to develop situational context the agent retrieves memory traces from these ontologies and compares them to the actual situation. If the perceptions from the actual situation match with the previous stored situational memories, situational context is recognized. By establishing situational context, the agent is able to produce speech thought statements in a certain scenario. The situational context is triggering an action, such as speech for example.

The result of the work is the generation of speech within a situational context. In order to produce speech a modular model from psycholinguistics has been integrated. The model is applied in a simplified form onto the ARS model to enable language ability (inner and outer speech) for the agents. The language is triggered through situation, which is by means of abstraction also able to generate speech in new situations, as it is looking for known structures in the old pattern. The abstraction mechanism is looking for matches in a new situation until something is found. If this is the case a known speech statement can be transfered as well to a new situation.

Kurzfassung

Die Arbeit löst die Frage der Errichtung von semantischem und situativen Kontext im ARS-Projekt. Kontext wird gebraucht, wenn zwei Agenten in einer bestimmten Domäne eine Frage lösen müssen. Er ist durch eine domain-spezifische Ontologie und eine allgemeine Ontologie festgelegt. Sprechen die Agenten über ein spezifisches Problem, ist es notwendig, dass Agenten die Problemdomäne kennen und spezifisches Wissen darüber haben. Lösen die Agenten ein allgemeines Problem wird weniger domain-spezifisches Wissen verlangt, allerdings mehr abstraktes Wissen. Aus diesem Grund existieren zwei verschiedene Arten von Wissensbasen.
Um den Situationskontext festzustellen, ruft der Agent Erinnerungen aus der Ontologie ab und vergleicht sie mit der aktuellen Situation. Wenn Empfindungen der aktuellen Situation mit den vorher gespeicherten übereinstimmen, dann wird ein Situationskontext hergestellt. Wird ein Situationskontext hergestellt, so ist der Agent in der Lage auf Basis der kontextuellen Hilfe zu navigieren und aus einer bestimmten Anzahl an Aktionen zu wählen, die aus dem Kontext kommen. Daher ist der Situationskontext ein Mechanismus für Aktionen, wie Sprache. Für Sprache kann semantischer Kontext zwischen zwei Agenten hergestellt werden, die miteinander kommunizieren.
Das Ergebnis der Arbeit ist die Generierung von Sprache aufgrund eines vorher hergstellten Situationskontext. Zur Sprachproduktion wird ein modulares Modell aus der Psycholingustik zur Sprachproduktion herangezogen. Dieses Modell wird in vereinfachter Form auf das ARS Modell übertragen, damit die Agenten eine Sprachfähigkeit (innere und äussere Sprache) haben. Es handelt sich also um eine durch die Situation gesteuerte Sprache, die durch Abstraktion auch in der Lage ist, in neuen Situationen Sprache zu produzieren indem bekannte Strukturen in neuen Mustern gefunden werden. Der Abstraktionsmechanismus sucht nach Übereinstimmungen solange bis auch in einer neuen Situation etwas bekanntes gefunden wird. Ist das der Fall, dann kann ein altbekanntes Sprachstatement auf eine neue Situation übertragen werden.

Acknowledgements

I am grateful to my supervisor Prof. Dietmar Dietrich for encouraging and supporting the work at hand during the last three years. As well as Dr. Klaus Dobelhammer and Mag. Kovacz Zsofia. Also, I owe much thanks to D.Oscar Corcho for making supportful comments and for being my second supervisor for the work at hand. Apart from this, I am grateful to my parents and my sister and my boyfriend for supporting me in the last year of this work.

Table of Contents

1 Introduction **1**
- 1.1 Overview of ARS . . . 1
- 1.2 Problem Statement and Motivation . . . 3
- 1.3 Methodology . . . 6
  - 1.3.1 Ontologies from past till now . . . 6
  - 1.3.2 Heading towards Psycholinguistics . . . 7
  - 1.3.3 Psychoanalytical Foundations . . . 9

2 State Of The Art **12**
- 2.1 Hypothesis . . . 12
- 2.2 Context and Context Aware Models . . . 12
  - 2.2.1 The Notion of Context . . . 13
  - 2.2.2 Context Aware Models using Ontologies . . . 14
- 2.3 Computational Models Using Knowledge . . . 20
  - 2.3.1 Computational Models Using Semantic Knowledge . . . 22
  - 2.3.2 Context Models using Semantic and Domain-Specific Knowledge . . . 27
  - 2.3.3 Speech in Cognitive Architectures . . . 28
- 2.4 Evaluation of Language Production Models for ARS . . . 29
- 2.5 Applied Fields Using Situational Context . . . 34

3 Concept **36**
- 3.1 Development of Use Cases . . . 36
- 3.2 Recognition of Situational Context in Scenarios . . . 40
- 3.3 Generation of Statements in Inner Speech . . . 43
- 3.4 Generation of Statements in Outer Speech . . . 45
- 3.5 Development of Listening to a Statement . . . 48
- 3.6 Overview of Data structures in the ARS model . . . 50
- 3.7 Development and Integration of new Data Structures . . . 52
  - 3.7.1 Data Structure Accoustics . . . 53
  - 3.7.2 Class Concept . . . 54
  - 3.7.3 Class Wording . . . 55
  - 3.7.4 Class Contextual Entity . . . 55
  - 3.7.5 Data Structure Action Executor . . . 56
  - 3.7.6 Assumptions for Linking the ARS Model to the Speech Production Model . 56
- 3.8 Organization of the Ontology . . . 57

3.8.1 Organization of the Context Ontology . . . 58
3.8.2 Organization of the Domain-Specific Ontology . . . 59
3.8.3 Memory Structure in the Ontology . . . 60

4 Modeling of Speech Production **63**
4.1 A simplified speech production model . . . 63
4.2 Concept Formation in Psychoanalysis . . . 68
4.2.1 Psychoanalytical View . . . 69
4.2.2 Neurocognitive Perspective of Modelling Speech and Language . . . 72

5 Design Methodology and Implementation **75**
5.1 Selection of Tools . . . 75
5.1.1 Tool for Simulation - MASON . . . 76
5.1.2 Tool for Ontology Engineering - Protégé . . . 77
5.1.3 Tool for Programming - Eclipse . . . 78
5.2 Realization of Situational Context . . . 78
5.2.1 Context Ontology and Context Entities . . . 79
5.2.2 Situational Context Ontology . . . 80
5.3 Realization of the Speech Production Model . . . 83
5.3.1 Semantic and Syntactic Speech Ability - F66 . . . 83
5.3.2 Decision Making - F26 and Reality Check on Actions - F53 . . . 85
5.3.3 Articulation Module and Motility Control - F30 . . . 86
5.3.4 Interfaces and Functions to other Modules . . . 88
5.4 Realization of Listening as an Reaction to Speech . . . 90
5.5 Realization of Alternative Actions . . . 92
5.6 Realization of the Use Cases . . . 92
5.6.1 Realization of Use Case 1 extended by Inner Speech and Outer Speech . . . 92
5.6.2 Realization of Use Case 2 extended by Listening . . . 93

6 Simulation Results and Discussion **95**
6.1 Simulation and Simulation Results . . . 95
6.1.1 Outline Overlay Icons . . . 95
6.1.2 Use Case: Situational Context . . . 97
6.1.3 Use Case: Inner Speech . . . 99
6.1.4 Use Case: Outer Speech . . . 101
6.1.5 Use Case: Listening to an Agent . . . 103
6.2 Situational Context: Ask and Listening . . . 104
6.3 Situational Context: Ask and Answer . . . 109
6.4 Discussion on the Implementation of Situational Context Recognition . . . 111
6.5 Discussion on Implementation of Inner and Outer Speech . . . 112
6.6 Discussion on Implementation of Listening and Speech Comprehension . . . 114

7 Conclusions and Outlook **116**
7.1 From Brave New World to Conceptualization . . . 116
7.2 Conclusions . . . 118
7.2.1 Conclusions for the Language Production Model . . . 119
7.2.2 Conclusions on the Selected Ontology Architecture . . . 120
7.2.3 Conclusions on the selected Use Case . . . 120
7.3 Further Work on the model . . . 121

7.3.1 Generation of Actions 121
7.3.2 The role of Outer Speech and Speech Comprehension 122
7.3.3 Future Fields of Appliance 124

Bibliography **126**

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
ARS	Autonomous Recognition System
BDI	Believe Desire Architecture
CEs	Context Entities
CONON	Context Ontology
DPs	Data parameters
ER	Entity Relationship
FOAF	Friend-of-a-Friend
GUI	General User Interface
MASON	Multi-Agent Simulator Of Neighborhoods
OWL	Web Ontology Language
PI	Perceived Image
RDF	Ressource Description Framework
SC	Situational Context
SOCAM	Service-Oriented Context Middleware
SOUPA	Standard Ontology for Ubiquitous and Pervasive Applications
TP	Thing Presentation
TPM	Thing Presentation Mesh
UC	Use Case
UML	Unified Modelling Language
WPM	Word Presentation Mesh
WP	Word Presentation
XML	Extended Markup Language

1 Introduction

"Knowledge is Power" James Bacon

Imagine an agent has to recognise a situation in a certain environment. Only when the agent knows the exact situation and the objects is it able to navigate in this novel situation. Human beings can capture semantic and situational context with relative ease, where agents or humanoid robots need either a lot of time or a lot of capacity. The notion of context as such has been explored by several scientific disciplines, ranging from computer science to cognitive [Her07, Vel08]. Consequently, the term context has many different connotations. The focus in this thesis is how context is applied in multi-agent systems for navigation and in speech for the purpose of communication.

1.1 Overview of ARS

Back in 2000, Dietrich and his colleague Sauter introduced the ideas of the Artificial Recognition System (ARS) model at a conference [DS00]. The goal was formed to model the human psyche, inspired by the second topographical model [Fre25, p.237] of psychoanalysis. It is a hierarchical, functional top-down model that has been developed since then, integrating the works of Zeilinger [Zei10], Lang[Lan10], Deutsch [Deu11] and Muchitsch [Muc13].

The functions in the ARS model are based on the Id, Ego, Super-ego model, also known as structural model or second topographical model[Fre25, p.236]. Psychoanalysis differentiates between conscious and unconscious content of the psyche. The functional modules are located in three layers. First, a neuronal network, second a neuro-symbolic network layer and third the psychoanalytical layer. Apart from a psychoanalytical approach and a neurological approach the project has used methods from neuro-psychoanalysis. Thus, Velik [Vel08] has developed a neuro-psychoanalytical model explaining the gap between the bottom neuro-symbolic network layer to the top psychoanalytical layer. Neuropsychoanalysis is a novel discipline that helps to bridge the gap between neurology and psychoanalysis.

Although different types of memory are used in the model of Velik, a loose definition of semantic memory predominates the model. This is simply, because the model does not focus on semantic memory concepts. Memory in the bionic model of Velik explains semantic knowledge and context knowledge [Vel08, p.104]. Semantic knowledge is considered basic knowledge of the world, whereas context knowledge influences perception.

However, the model has grown rapidly from a few functions to a vast number of forty different functions, by means of the top-down approach that was chosen. Fig.1.1 shows the current version of the ARS model 1[1]. According to psychoanalysis, one can differentiate between unconscious and conscious processes [Fre15b, p.172]. Functional units belong to the primary processes work unconsciously, whereas functional units belongto the secondary processes are conscious.

Fig.1.1 is a very simplified version of the processing cycle of the agent's psyche in the current version of the model. At the beginning, external perceptions from the Environment are processed in the Perception Track. Secondly, perceptions are stored and processed in the Memory Traces for Perceptions module. Thirdly, perceptions are processed by rules from the Defence apparatus. In order to get to the secondary process, a Transformation has to occur between the primary and the secondary process. In the Decision Making and Generation of Imaginary Actions modules, the actual word selection happens. Finally, the Motility Control and Actuator Output modules are responsible for the word production as an output to the environment.

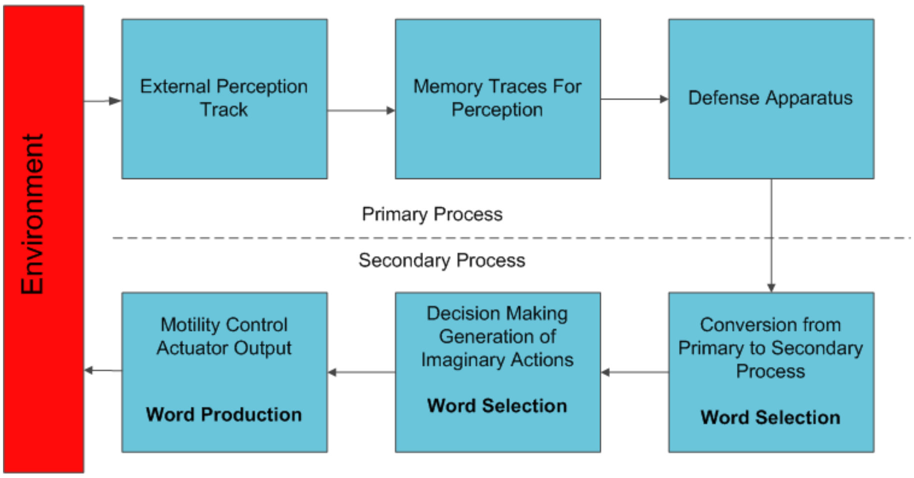

Figure 1.1: Very Simplified Version of the ARS Model

Project Environment

The section explains how the approach of the ARS model has been integrated theoretically and practically in existing projects. To illustrate this, two proposals ENTER and THINKHOME [2] are described in the following. Practically, distinct components of the current ARS project can be employed to optimise energy efficiency in building automation systems. This optimisation can be realised through a bionic approach of simulating context awareness, as well as domain-specific and general memory organisation. Exactly this context awareness has been addressed in the THINKHOME project [RKIK11]. For example, in building automation systems and implementations, the system perceives a vast amount of sensory input perceptions through various sensory nodes. Thanks to these sensory nodes, a lot of information comes together, resulting in a sensor fusion. Thus, current building automation systems capture a vast amount of sensory data, which has to be evaluated according to user needs. One problem caused by this richness of data, is context awareness for inhabitants living in the house.

[1] Link provided at: http://www.arsini.org/wiki/doku.php?id=model:v4.v38:functional:overview

[2] Project has been sponsored by FFG Haus der Zukunft Plus P822 170.

The problem can be solved by implementing the recognition of situational context through usage of an ontology [3]. In the project, habits regarding heating and cooling are monitored continuously and stored in a domain-specific and general ontology. In addition, general facts are stored in a general, abstract ontology. By using two different ontologies, the situational context of a specific user can be captured and an action is set by the intelligent home environment. The action in the cases of the THINKHOME project is reducing heat or light conditions in a room, depending on the inhabitants in the room.

Apart from optimisation of energy, the ontological component of the ARS model can be applied in the field of robotics for situated awareness. For instance, a robot has to enter a kitchen, because it has the task of preparing tea. To fulfil this task successfully, a lot of knowledge is required, such as domain-specific and abstract knowledge, e.g. knowing what distinct objects are in a kitchen domain and what is generally expected in a kitchen. The problem is the vast amount of different knowledge that is required for fulfilling the task successfully. Different kinds of knowledge are required to recognise any kitchen that may present itself. Meaning every time the agent enters a new kitchen, it will find the same items there.

The problem can be solved by adopting a multi-layered architecture of several ontologies. The proposed architecture contains four different ontologies. One that is used for recognising objects, one for affordances of the objects, one for processes that can be conducted and one for tasks that can be fulfilled. The architecture of the ontological architecture goes from lower to higher processes, meaning a range of domain-specific to general and abstract knowledge is required. The proposal ENTER [BVH11] is based on underlying research at the ACIN (Institute of Automation and Control) department, which works on situated vision and grasping of items in combination with a robotic arm.

This thesis is inspired by the framework of these two proposals. The key aspect in the THINKHOME project is selecting the appropriate ontology for a user-centred smart home scenario. The central aspect in the second project proposal ENTER is developing a semantic ontology in combination with situated vision. The project deals with situational awareness in computer vision and the design of a semantic ontology. This ontology is composed of several layers of object, process, affordance and semantic memories. Gibson states that the affordances of the environment are what it offers, what it provides or furnishes [Gib79]. In short, affordances represent the expectations one has to an object, an animal or a person.

Inspired by these two projects, the motivation is to develop a user-centered ontology for the detection of semantic situational context. In the next step, the semantic and situational context defines the framework for speech and thoughts that are based on a speech production model. The underlying methodology [4] is based on computer technology, experimental psycholinguistics and psychoanalysis.

1.2 Problem Statement and Motivation

When human beings enter a scenario for the first time, they have the ability to orientate themselves in a new room or situation based on fixed concepts or templates of their environment. What human beings do, is an abstraction before they focus on a domain-specific part in their environment. This

[3] An ontology in the ARS project is a knowledge base storing domain-specific and general, abstract information

[4] A methodology is used here as a guideline for solving a problem with specific methods.

seemingly easy orientation task becomes increasing important in scenarios where agents navigate in a human-like world, e.g. in the ARS model. Agents in the ARS model, called ARSINs, have a considerable problem with this task. They are not able to orientate themselves in new surroundings based on contextual aids. This is due to their inability to capture the situational context of a situation. Through recognising situational context in the ARS model, context can be captured and situational awareness is gained by the agents.

In addition, situational context recognition is a valuable aid in orientation tasks, when stored object configurations and actual perceptions are compared to each other.

This kind of situational awareness is a valuable aid in communication situations. Therefore, by means of situational and semantic context, awareness agents in the ARS model can communicate more naturally. The problem that an agent's communication lacks knowledge of concepts, is solved in this thesis by establishing a conceptual basis first and an ontological basis second. The conceptual ontology is accessed in the primary process, whereas the ontological basis is accessed in the secondary process. Added to that, domain-specific and general knowledge contributes to situational context finding in communication scenarios. Meaning, agents know how to act appropriately regarding their speech statements and actions.

As long as situational context is not developed, the agent cannot establish a situation awareness that is needed to understand the meaning of a situation. Furthermore, situational context precedes language production. Meaning without knowing a situation, language can not develop and can not be generated. In addition, in new situations a greedy search mechanism looks for known information until it finds something connected to the old known knowledge.

Task

The specific task is to anchor situational, semantic context based on a conceptual design and an ontology as a knowledge storage.

Therefore, four steps are needed:

a) to recognise situational context,

b) to generate the concept,

c) to interpret and assess the meaning of an object,

d) to enable storage in form of memories in the ontology.

The agent has to recognise the configuration of several objects and subjects in a certain environment in order to achieve higher situational awareness. The environment at hand is the simulator engine. For situational awareness, however, not only entities, but situations are compared to each other. The advantage of generating such a context for situations, is that several situations can be recognised based on different context situations. The context situation is stored in a domain-specific ontological structure. Although the variability of objects is not very high in the model, the variability in situations has increased due to different use cases that have been designed for the evaluation of the psychoanalytical content.

Currently, three different use cases exist [DBM+13]. A use case can take different directions based on variable situations. Thus, different sub-use cases exist and the variability of actions is quite high, which means that, based on a certain setting, different actions can be conducted by an agent. At the moment, alternative actions relating to a context situation are not part of the ARS model. That is what situational context awareness changes; different situational contexts can trigger alternative behaviour in the agents.

In order to know the situational context, the existing configuration of objects has to be compared to a previously stored version in the memory. This requires access to memories about situations and conducted actions and an interchange with the reality check. A filter criterion can be, for example, that actions that occur more often are retrieved with a higher probability than situations that occur less often. If a new situational context is recognised that has not been stored in the ontology yet, it cannot be learned by the model, which means that new situations, if they are not part of the ontology, can be stored only manually in the underlying ontology.

For example, Bodo and Bella are agents in the ARS model and simulation. They are in the house standing in front of the cake. Bodo remembers a situation with Bella and the cake near to him. In this situation, he was hungry and thus he approached the cake to eat it on his own. However, in the current situation he is not hungry. He compares the actual situation with another situation from his memory where he was not hungry and his reaction was to wait until the other agent eats the cake. Based on the situational context here where his hunger level is low, Bodo waits until Bella eats the cake first.

In addition, in the ARS project agents need the ability to comprehend the meaning of objects by means of their occurence in a situation. It means foremost that agents need the ability to comprehend the meaning of situations. From the situation they know how to treat an object. E.g. a cake in the house of another agent implicates that the cake is owned by this agent. Object classification is part of the model in a simplified version, where one configuration is compared to another. In the old version of the model, the meaning of an object cannot be accessed and comprehended by the situational context whereas in the new model proposed here it is possible to access the meaning via the situation.

Situational context (b) is always domain-specific based on abstract concepts. Abstract concepts are part of the general, upper ontology, which is described in detail in Chapter *Concept*. At first, the agent is navigating based on a general concept, which is mainly an abstraction. As long as no situational context is recognised the situation is abstract. Situational context is recognixed if each of the four concept entities are filled with domain-specific attributes. These are filled during a transformation phase, which is a transformation from the acoustic Thing presentation mesh (TPM) to a WPM (word presentation mesh) and in a second step the extraction of a concept from a WPM.

Situational context is generated based on the recognised concept entities and domain-specific attributes attached to them, e.g. the Concept Entity Action: Flee. In a domain-specific situation where a speech or thought action is needed, only one domain-specific attribute is retrieved. In case a speech statement is required, the action entity is needed. Situational context is therefore used as a trigger for thought and speech statements.

For object interpretation (c), it is required to relate an acoustic thing representation to the visual pattern of the word representation in order to capture the meaning of the object. It is needed for establishing a relation between the meaning of a word and its meaningless representation. Thus, a pointer needs to be generated from the meaningless acoustic representation to the actual word presentation.

Situational context (d) and concept knowledge needs to be stored so that it can be accessed later. Thus, storage of the memories is realised in an ontology. The ontology is a knowledge base that contains memories of the agents that the agent has experiences and on the other hand the concept ontology entails the concept by means of object descriptions.

The final goal is that the agent recognises the meaning of a situation based on the actual perceived environment, memorised concepts and the association between actual perception and related memorised concepts of the domain-specific situation. Situational context is anchored if the agent approaches another agent, it is linked to a rise in the affect.

The actual configuration of entities of the situational context is compared to the stored configurations in the ontology. When a scenario is new, it is not used in one of the next scenarios as situational context. In other words, automatic context recognition or the learning of new contextual situations is not planned for the work at hand. Thus, situational context will only be recognised based on manually stored configurations in the ontology. For extending stored configurations, one needs to extend the current ontology. As regards the stored content, all involved entities are stored in addition with the pleasure value of the situational context.

1.3 Methodology

This section sums up all methods needed in order to realise semantic and situational context in the ARS model. The development of context is an interdisciplinary phenomenon which takes place in three different fields. Computer science and cognitive sciences have discussed the impact and development of semantic, situational context. To achieve situational context in a computational model, three disciplines are needed: Computer sciences, which is responsible for the ontological design; psycholinguistics, which is a discipline of experimental psychology that has contributed to the discussion of context; as well as psychoanalysis, which offers contribution ideas due to "Deckerinnerungen" that engrave emotional situations in the past.

1.3.1 Ontologies from past till now

The expression *ontology* stems from the Greek words ontos, meaning being, and logos, meaning science. It was originally founded in the 17th century and was defined as the "science of the being". In the category science of Aristotle, the philosophy contributes to the existence of the ontology. In his category science, Aristotle aims at the classification of expressions. He asks the question, how categorical expressions are connected semantically to each other. Characteristic for his work is that logics and semantics are not separated from each other. Semantics aims at accessing the meaning of items or objects in the world. Nowadays it is most common to separate the semantics from the logics, as we can see in the formation of language. Nearly all spoken languages can be split into syntax and semantics. This dualism is also represented in programming languages, where we can find syntax - the definition of all expression that can be built within the language, and semantics.

"An ontology is an explicit specification of a conceptualization" [Gru93, p.2]. In computer science, ontologies are used in order to establish a formal definition of things, concepts and relations. Ontologies can be seen as classifications that can contain rules. Due to the complexity of the rules, ontologies are seldom used. The reason for this is the possibility to separate rules from the ontology in the programming logics. In the ontology one can find specific knowledge about a domain. Thus, knowledge in an ontology is defined by one or more contexts, as context is generated within a domain. For example, working in the health sector involves different domain knowledge than that for working in the banking field.

Context is set by the domain we live in. A doctor in a hospital communicates in a certain domain. The expression context defines in what knowledge domain (domain ontology) one is set [Her07, p.144]. Therefore, context defines which knowledge is required for working and communicating in a specific domain of context. Someone who is not familiar with the context in a situation, is not able to react fast in a specific situation. This is due to not having enough knowledge about the ontological knowledge of a community. A concept is an abstraction that is derived from a specific instance within a domain.

The ARS (Artificial Recognition System) has a need for semantics for multiple reasons:

- Enabling storage of knowledge,
- Semantics for object conceptualisation,
- Enabling semantic context information.

Knowledge of an agent is stored in an ontological way. Semantic knowledge in ARS resembles the general knowledge of the agent, such as how the world is built, the properties of the objects and subjects in the world. Semantic knowledge is a kind of metaknowledge on how to describe the world. Semantic knowledge also includes the use of language and the production of language as well. It is in contrast to episodic knowledge, which contains the individual knowledge of an agent.

Semantics in ARS is needed when the agents understand the meaning of an object based on drives in the primary process. Based on semantics - the meaning - agents are further able to conceptualise objects in the secondary process. Conceptualising means to build an abstract concept by which we can classify new or different perceptions. Without understanding the meaning of a concept, one cannot subsume sub-concepts.

Situational context information is needed for orientation and localisation purposes in the world of the agent. Semantic context encompasses the objects and subjects in a certain situation, linking them together through a common meaning. All entities in a given semantic context are linked through the same contextual meaning. For instance, if the agent is in a certain situational context, it can trigger a plan, action, event or memory. Different interpretations of contexts may lead to different actions.

1.3.2 Heading towards Psycholinguistics

Language can be seen as a means of communication for telling concepts and feelings to another person. This assumption is based on the meta model for language production of W. Levelt [Lev89, p.9]. In his prior work regarding speech production, Levelt describes the speaker as a highly complex information processor who has the ability to transfer intentions and feelings into language. Levelt assumes a functional model that executes several algorithms. In addition, he distinguishes between microplanning and macroplanning. Macroplanning deals with searching the memory content, inferences and decisions for subsequently communicating something.

The thesis, however, focuses on the level of microplanning, which decides how something is finalised so that it can be communicated in the end. Microplanning is connected to context in such a way that context is influenced by the concepts and intentions for speech. The awareness of situations is interrelated with the ability to comprehend concepts. A concept is an abstraction of characteristic properties that represents something in a general way. A situation can only be comprehended if the concepts in the configuration are known. Thus, understanding a situation requires knowing and understanding of all existing underlying concepts.

Speech per se is not sufficient for realising semantic context. The model of Levelt offers a good explanation for embedding context by means of concepts and intention and has therefore been chosen. However, in *Chapter State of the ART* it will be explained in more detail why the speech production model of Levelt fits well in the ARS project.

The speech production model is composed of the following modules: |

- the Conceptualiser unit,
- the Formaliser unit,
- the Articulator unit.

The *conceptualiser* forms the speech intention of the speaker, for which it needs ontological access, such as conceptual and situational knowledge. The conceptualiser contains a monitor and a structure for message generation. The monitor is a structure that parses the generated message (the speech) if it equals the chosen concept. The message generator turns the intention of the speaker into a preverbal message that is sent to the next unit. The preverbal message has semantic character, as context knowledge and situational knowledge are integrated at this point.

The *formulator* receives the preverbal message as an input and generates the phonetic plan, which is sent to the articulator as the last unit before speech generation. A syntactic buffer is included in order to retain more complex phonetic plans that need to be active for some time.

The result of the phonologic encoding and planning are articulatory plans that are internal representations of the phonetic articulation. These plans are sent to the articulator in the last stage. In the articulator, the plans are processed. In addition, an articulatory buffer exists, where plans are active for some time.

Fig.1.2 below shows how the three-layered ARS model will be integrated into to speech production model. The psychoanalytical layer entails all the functions subsummarizing speech, such as the conceptualizer unit, the formaliser unit and the articulator unit. Meaning, the psychoanalytical layer contains all functional modules for speech. These modules will be described in a greater detail in Chapter Modeling of Speech Production.

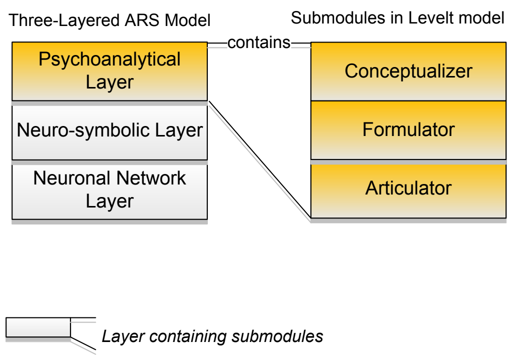

Figure 1.2: Integration of the Speech Production Model in the Psychoanalytical Model

Quite important for the model is that Levelt distinguishes between the conscious state and the unconscious state as well. He assumes that most of the processes are unconscious, which is in contrast to psychoanalysis. The phonetic plan resembles the so-called internal speech. Inner speech or internal speech means to look at objects (not necessary) and speak in thoughts about them. Finally, it needs to be emphasised that Levelt favours the spreading activation theory of Dell [Del86], because it is sufficient for the theoretical requirements of the speech production model. The spreading activation meachanism has been integrated in the ARS model [DBM+13], which uses spread activation for generating association of the agents, for example.

1.3.3 Psychoanalytical Foundations

Psychoanalysis is fundamental in the ARS model, as it has inspired the growing and development of various functional modules and the whole model as such. The second topographical model of Freud dates back to 1923 [Fre25, p.237], where it was first mentioned in literature. The concept of the word- and thing presentation however, was first described in 1891 in the studies of aphasia [Fre91] and then in the theory of the unconscious. In his theory, Freud separated the object representation into the word presentation and the thing presentation, which he described as psychic content cathected with quota of affect.

Freud explained his theory on language, reading and writing in his work about aphasias. He states that the word presentation acquires its meaning when connected to the object association, at least for substantives. The object association is an association complex consisting of different visual, acoustic, tactile and kinaesthetic representations. The object representation reflects a thing that can be perceived by these different modes of perception (visual, acoustic, tactile and kinaesthetic). The word presentation is linked to the object presentation via the acoustic patterns [Fre91, p.121].

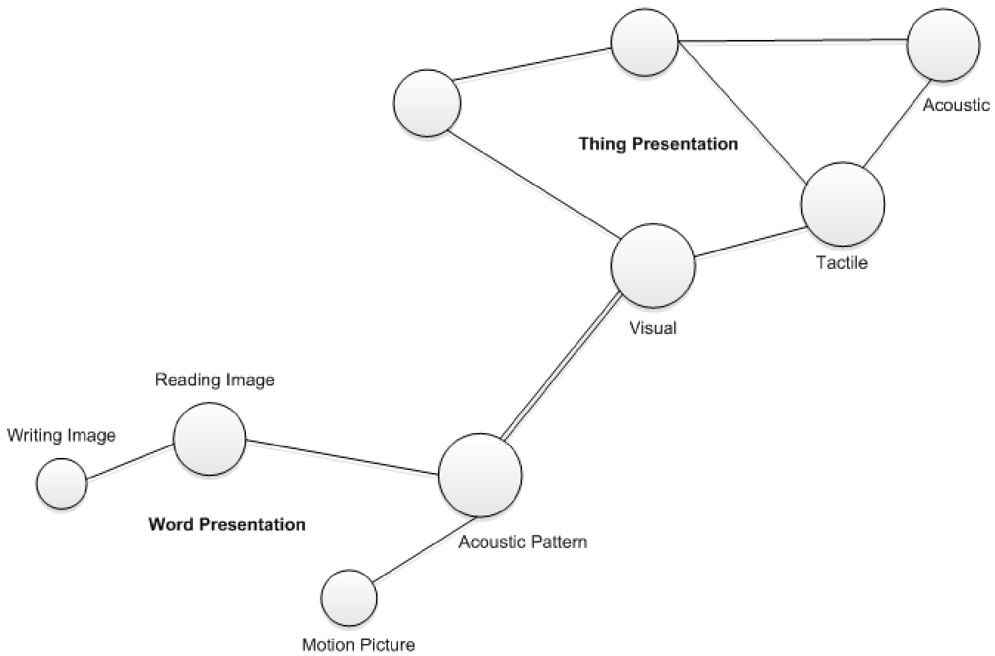

Figure 1.3: Description of Word Presentation Linked to a Thing Presention [Fre91, p.121]

Fig.1.4 describes the transfer from unconsciousness to preconsciousness first, where in first step thing presentations are connected to word presentations. In the next figure the link between a thing presentation and word presentation is cathected by means of psychic energy. According to Freud a presentation can become conscious only if the thing presentation is linked to the matching word presentation. Moreover, the amount of the cathexis decides if a content is in a conscious state or remains in a preconscious state. In schizophrenia words are linked to the same process that makes out of latent dream thoughts latent dream pictures. Meaning in schizophrenia word representations are processed according to the same process as thing presentations in the primary process [Fre15b].

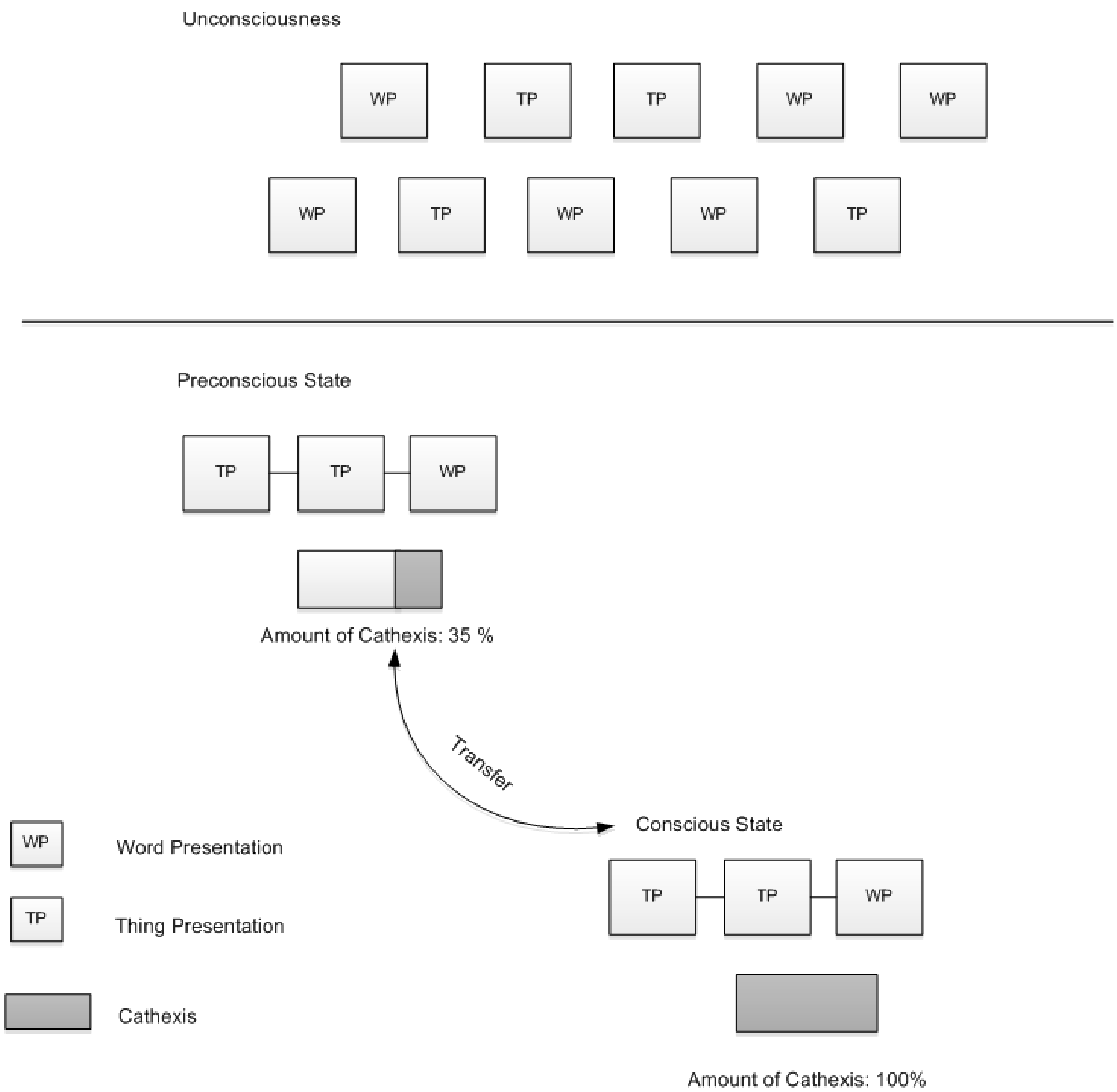

Figure 1.4: Transfer From Unconscious to Preconscious and Conscious State

Crucial is the fact that the word presentation is situated in the secondary process and the thing presentation in the primary process Thus, the concept of a word presentation in Freud's theory is connected to the speech production theory as well. Word presentations as such are needed when semantic context in language is defined. In order to establish situational context, only abstract concepts that are derived from object and thing associations are needed. Consequently, situational context is situated in the unconscious part of the psyche, whereas semantic context in language is located in the conscious part.

Finally, to develop semantic context in language, this work assumes the prerequisites of situational context and the foundations of word presentations to realise semantic context. Fig.1.5 describes how the thesis has been planned in a hierarchy. First, the concept is to develop situational context for navigation and orientation purposes. Second, the foundation of language in psychoanalysis requires word presentations. These are so-called units of speech. Last, semantic context is established from the two preceding steps. Depending on a certain situational context and the prior selection of a word presentation, a language statement can be produced. Further details of each of the functions can be found in Chapter *Concept and Model*.

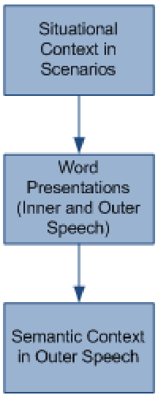

Figure 1.5: Illustration how Semantic Context in Language is Generated

2 State Of The Art

"The next best thing to knowing something is knowing where to find it. " Samuel Johnson

After discussing how context is defined, some computational models of context are brought forward at the beginning of this chapter. The generation of context in context aware models using ontologies is discussed. Furthermore, it is described how domain-specific and semantic knowledge is used in computational models for the generation of situational context. As described in the last chapter, situational context is implemented by means of a language production model. Thus, various language models are presented, their features are discussed and an evaluation based on the ARS model is done at the end. Finally, context is applied to building automation and relevant projects in the field are reviewed.

2.1 Hypothesis

The hypothesis laid out in the work is the simulation of context generated inner and outer speech in the Artificial Recognition System (ARS). Inner and outer speech are based on a situational context frame. Thus, in the first part of the State-of-the-Art, models that are implementing context are introduced further. In order to establish semantic context a semantic ontology is used. An overview of models using semantic knowledge is given starting from connectionist models that implement semantic knowledge. The third interdisciplinary aspect in this work is the evolving of speech and language among the agents. Thus, speech production models are discussed and evaluated and in the end a model is chosen that is most suitable for the ARS model. However, the final aim is to implement inner speech as thoughts and outer speech as articulated words in the ARS model.

2.2 Context and Context Aware Models

The following chapter discusses the notion of context including a description for situation-awareness and context-awareness. Then, four different context aware models using context are discussed, ranging from the CONON ontology, Cobra, Situation to the SOCAM ontology.

2.2.1 The Notion of Context

At first, one has to define the notion of context. According to Albrecht, context is "discourse that surrounds a language unit and helps to determine its interpretation, the set of facts or circumstances that surround a situation or event [Sch00]". Depending on the field of discourse that surrounds the unit, different types of contexts can occur. According to Galleguillos and Belongie [GB08], three different types of contexts exist in computer vision:

- Semantic Context,
- Spatial Context, and
- Scale Context.

Further, they claim that semantic context is the probability that an object occurs in a certain semantic scene. Spatial context is the probability that an object occurs in a certain position with respect to other objects in the surroundings. Scale context is the probability that an object occurs relative to a certain scale, such as a temporal or metrical scale. The definition is related to the field of computer vision [GB08].

However, definitions of context exist in other fields than computer vision, as well.

According to Dourish, context can be seen as a representational problem or as an interactional problem. First, context as a representational problem is more related to the positivist's theory of context being regarded as something quantitative from natural sciences. The representational view describes context as a problem of representation, similar to the case in computer vision.

Secondly, context as an interactional problem entails the sociological views. Context is stable when it is mutually recognised by the parties. Apart from this, it covers the topic of context in a conversation. Context in a conversation is represented if two people share a history of interaction and some shared knowledge about the topic is considered contextual information.

Thus, context can be modelled as representational problem and as interactional problem. The thesis at hand deals with both approaches. Situational context addresses the representational view, whereas semantic context in speech contributes to the interactional approach.

Context Awareness

Developing context in pervasive computing systems [5] is a challenge in the field of context aware computing (CAW). The field of CAW is subsumed by situation awareness (SAW). To achieve SAW, the relevant relations among objects have to be identified. Therefore, it is required that the meaning of these objects can be comprehended [BR07]. The comprehension of meaning for objects and situations is crucial for the implementation of speech in ARS. That is why in this work the agents aim to understand the meaning for objects and situations to comprehend situational context.

Schilit and Theimer regard context as location, identities of nearby people and objects, and changes to those objects [ST94]. The definition applied onto the ARS model refers to the environment in which agents and objects are located and changes of the environment. Changes in the environment are triggered from context in this case.

[5] Pervasive computing is part of the field of ubiquitous computing.

Dey defines context in his work as "any information that can be used to characterize the situation of entities" [Dey00]. Context information is used to describe a situation of objects for an agent. The approach can be used in the ARS model, if all available context information describes the situation for an agent. The term situation as an abstraction is underlined in the work of Dey and can be seen as "the states of relevant entities" [Dey01, p.8].

In the thesis at hand, situational context is attributed to the field of SAW, because relevant relations among objects are indentified. Meaning, situational context in the work at hand is determined by the five context entities. A detailed description of the relation of the concept entities is available in chapter *Concept*.

2.2.2 Context Aware Models using Ontologies

The following subchapter focuses on models that develop context and use an ontology-based approach and apply it.

First, the CONON model is introduced [WGDP04]. The upper ontology in fig.2.1 shows the OWL program classes for the items *Context Entity, Computational Entity, Location, Person and Activity*. The central class is the entity *Context Entity* and Location, Person, Activity and Computational Entity are derived. OWL classes [7] are referred by two different kinds of arrows. Namely, rdfs:subclasses and OWL:properties. *Service, Application, Decice, Network and Agent* a[re rdfs:subclasses [8] of the class *Computational Entity*. *Location, Person and Activity* are also described by rdfs:subclass of relations. In the domain-specific ontology, OWL classes are built up by rdfs:subclasses and OWL:properties. In addition, the domain-specific ontology can have diverse domains, such as the *Home-Domain* and the *Office-Domain*. Context reasoning is used for checking the consistency of context and abstracting high-level context from low-level context.

[7] OWL classes are program classes from the semantic meta language Web Ontology Language.
[8] RDFS stands for Resource Description Framework.

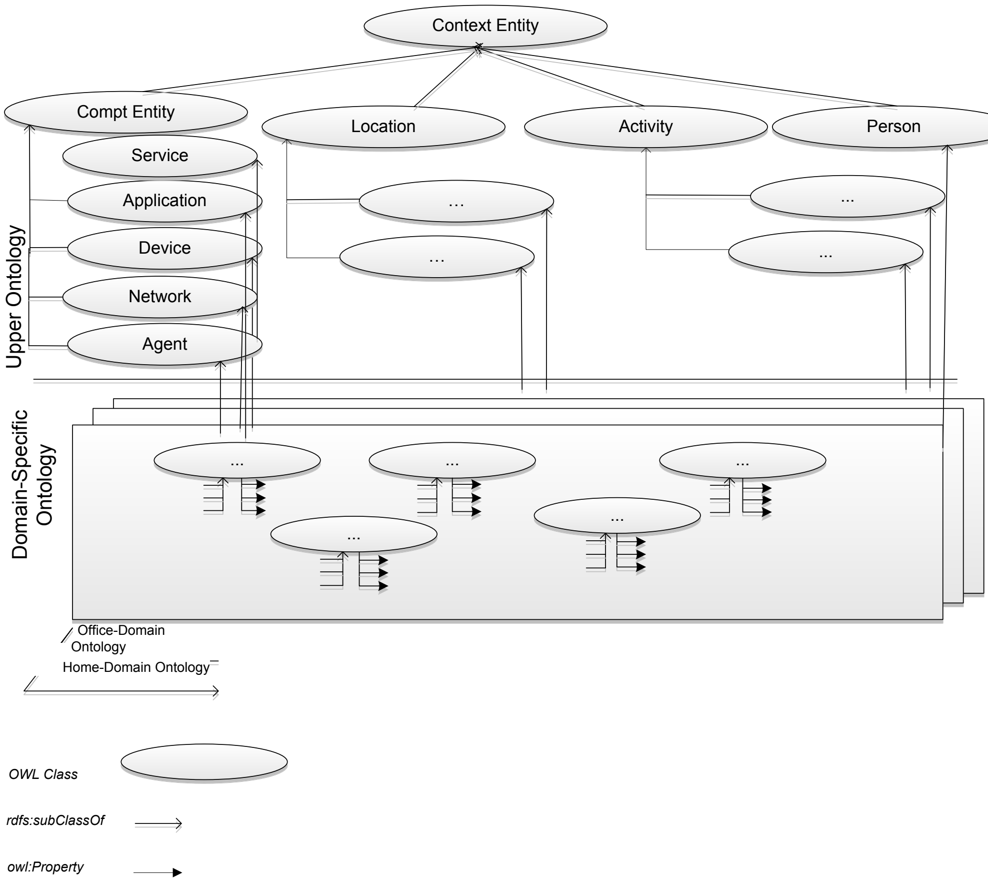

Figure 2.1: Conon Ontology [WGDP04]

The model has been founded by Wang and his colleagues [WGDP04] Semantic Web ontologies and technologies are applied in the field of context modelling and context reasoning. The model is based on an upper ontology that integrates context by means of the Context Entity and a domain-specific ontology. General, upper ontologies are used for integrating information and sharing knowledge among heterogeneous sources [BR07]. The aim of the context model is to define upper level components that can be extended to several specific domains. This is based on the fact that applications are grounded in pervasive computing environments as sub-domains. They propose that *Location, User, Activity and the Computational Entity* are the major context components in the context model. Context in each domain shares common concepts and can therefore be modelled using a general context model.

A separation into an upper ontology and a domain-specific can be done regarding the knowledge base in ARS. The upper ontology in ARS reflects the main context entities such as Action, Distance, Enitity, Emotion and Drive, that are used to identify situational context, whereby the domain-specific ontology can be used to identify situational context on a domain-specific level. Both levels however are needed for correct identification. At first via the upper ontology categories and secondly it becomes more detailled in the domain-specific architecture. The advantage of this structure is a clear separation between high level, more abstract and low level, domain-specific ontologies.

Fig.2.2 represents the Situation Ontology by Yau and his colleagues [YL06]. The ontology models context and situations by developing a general upper ontology with concepts and relations. The knowledge representation language used is OWL DL [6].

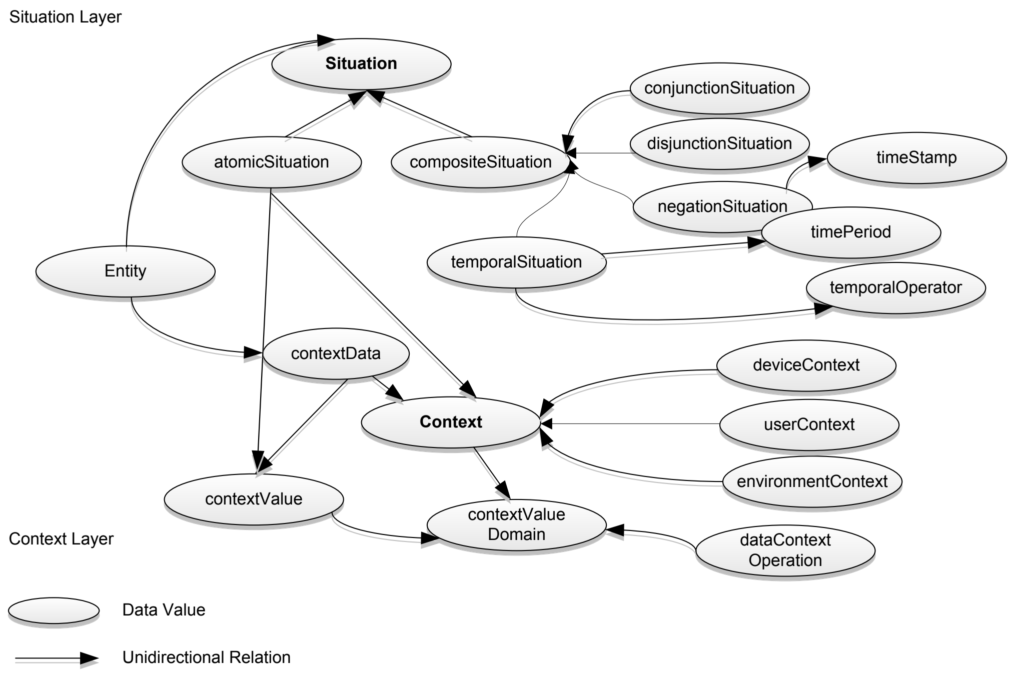

Figure 2.2: Illustration of Situation Ontology Using Context [YL06]

The general upper ontology can be extended by domain-specific classes as well. The ontology separates between a context layer and a situation layer, whereby the context layer stands for the context definition. Context is defined by different types of context, such as *deviceContext*, *userContext* or *environmentContext*. Further, *contextData* is part of the *Context* and each *contextData* has a *contextValue*, a time stamp and an ID. The context in the upper ontology has a domain. Further, context belongs to a *contextValueDomain*, that has a *dataContextOperation*.

The situation layer subsumes context into situations and develops complicated situations out of simple situations. The *situation* is composed of an atomicSituation and a *compositeSituation*. Later one consists of a *temporalSituation* having a *timePeriod* and a *temporalOperator*. Furthermore, a negationSituation, a disjunctionSituation or a conjunctionSituation. The negationSituation also has a timeStamp and is part of the *compositeSituation*. The two layers *Situation* and *Context* are connected by the *atomicSituation*, where each atomic situation has a context. Another link is established by the *Entity* that is related to *contexData* and satisfies a *situation*. The ontology uses reasoning for consistency check, subsumption reasoning and implicit knowledge reasoning. In contrary to the CONON ontology, the advantage of this ontology is the representation of relations as well. Apart from this the whole ontology can be seen as an upper ontology. This means of representation is less suitable for the ARS model due to the fact that the identification of situational context is not homogenious.

SOUPA is the abbreviation for Standard Ontology for Ubiquitous and Pervasive Applications, founded by Chen and his colleagues [CFJ05]. It is based on a Context Broker Architecture

[6] DL stands for Description Logics, which is a formal knowledge representation language

and is split up into two ontologies - SOUPA Core and SOUPA Extension. SOUPA Core has been developed for context awareness, whereas SOUPA Extension supports applications, such as various other ontologies including OpenCyc, Friend-Of-A-Friend ontology (FOAF), MoGATU BDI ontology. Open Cyc and the MoGATU BDI ontology include context. The core ontology includes the OWL classes *Person, Policy, Action, Agent, Time, Space and Event.* The SOUPA extension includes owl classes, such as *Document, Meeting, ImgCapture, Digital-Doc, Schedule, Location, Region and Device.* The bracktes indicate the XML namespace that is used and the arrows imply subclasses by using owl:imports. The SOUPA ontology is presented in fig.2.3.

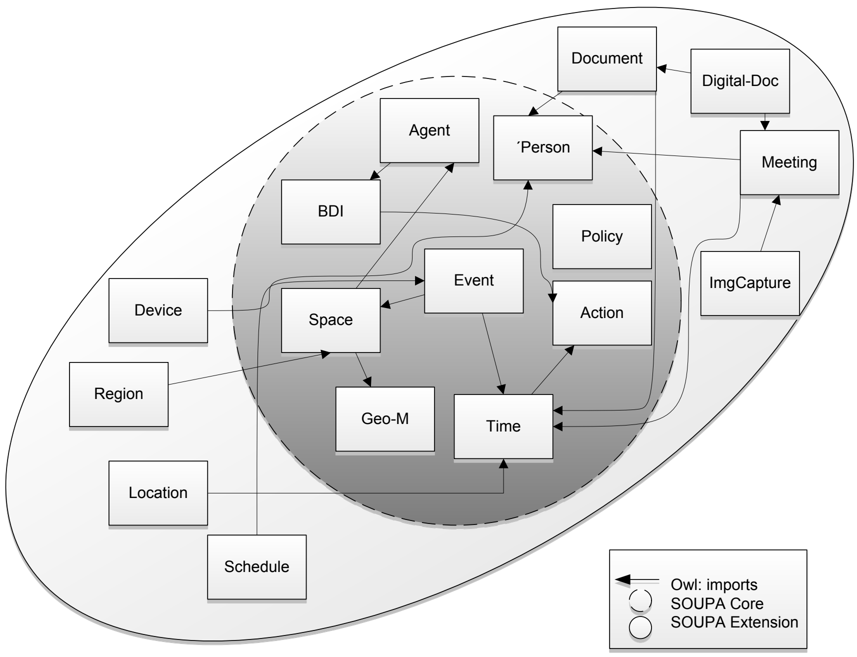

Figure 2.3: SOUPA Core and Extension Ontology [CFJ05]

The SOUPA ontology represents a quite clear example of separating two different ontologies. On the one hand SOUPA core captures the most important items that are needed in the ontology to capture context. On the other hand the SOUPA extension ontology includes additional applications that are not necessarily required all. The structure of the SOUPA ontology can be transferred onto ARS ontological structure in the following. Meaning, the core ontology resembles classes that do not change and are required at least. These are the functional modules for decision making, for example. The extension ontology reflects classes that are not needed with every start of the ontology, which would be access to domain-specific parameters.

SOCAM (Service-Oriented Context-Aware Middleware) [THD04], see fig.2.4, is an architecture that makes use of perceived context information based on Simple Object Access Protocol (SOAP).

The architecture consists of external sensors and internal providers based on SOUPA that request information from the sensors. It is modeled in Web Ontology Language (OWL) and contains the *ContextEntity, Computer Entity, Location, Activity and Person.*

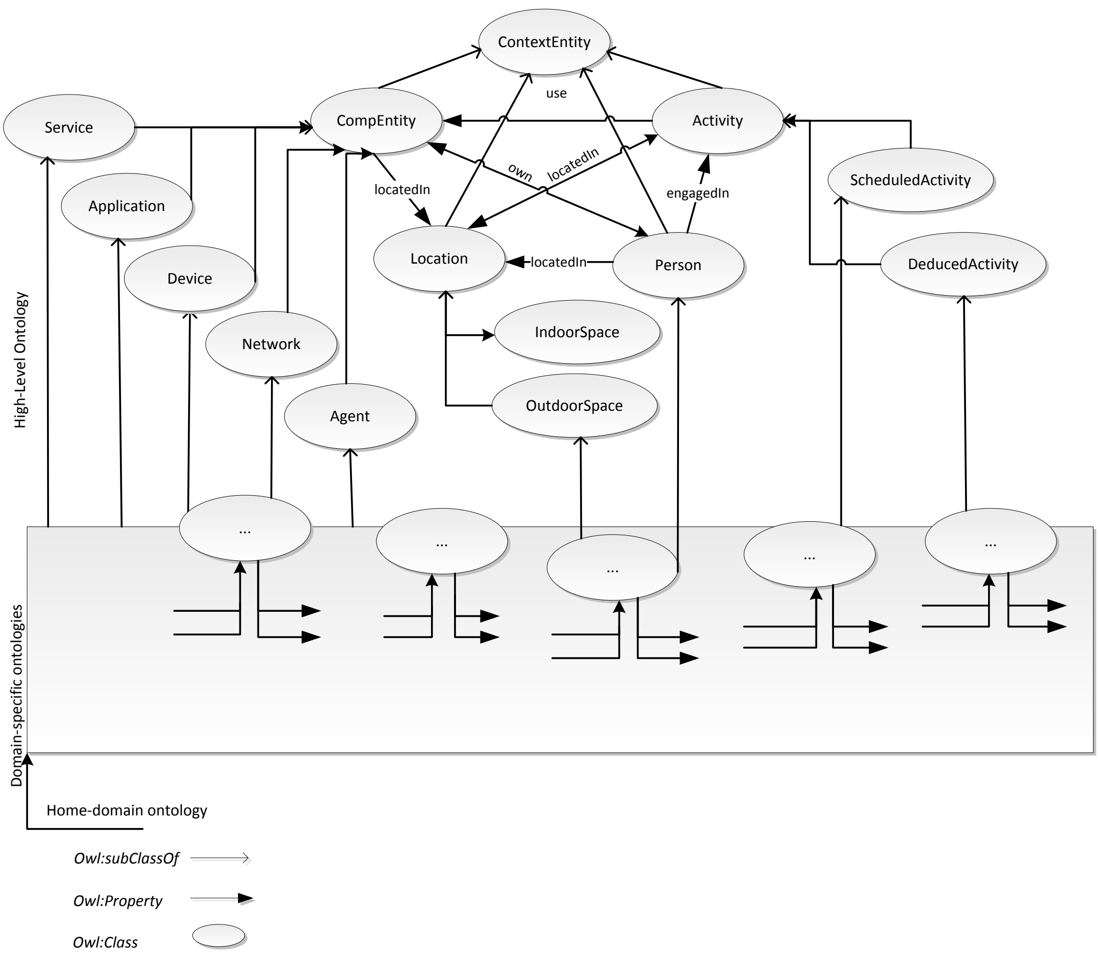

Figure 2.4: Structure of the SOCAM Ontology [THD04]

In the following description the fig.2.4 is explained. The *Comp Entity is located* in a *Loaction.* Also an *Activity* and *Person* are located in a *Location.* A *Person* is *engaged* in a *Activity.* An *Computer Entity* is *owned*by *Person.* A *Computer Entity* is *used* for an *Activity.* In general, the *Computer Entity* is broken up into the subclasses. *Service, Application, Device, Network and Agent. Location* is broken up into *IndoorSpcae* and *OutdoorSpace. Activity* is broken up into *ScheduledActivity* and *DeducedActivity.* In between the five main nodes there are either rdfs:subclassof or OWL:properties, whereby features are represented as OWL classes and feature relations as OWL properties. The rdfs:subClassOf is a part of OWL ontology reasoning rules and is defined in a formal way. SOCAM distinguishes between a general upper ontology, modelling the later instances, and domain-specific ontologies that can be defined. Based on the context data, learning algorithms enable a high context awareness [THD04]. SOCAM ontology is a perfect example for a general upper ontology and a domain specific ontology. In addition to the CONON ontology, SOCAM introduces upper ontology items that are linked to each other by associations. Another difference is that upper ontology items can be split up more detailled. The consequence is a more detailled upper ontology in this case. For the ARS model this type of ontology would be suitable. However, the identification of situational context does dependend on four principle contextual entities.

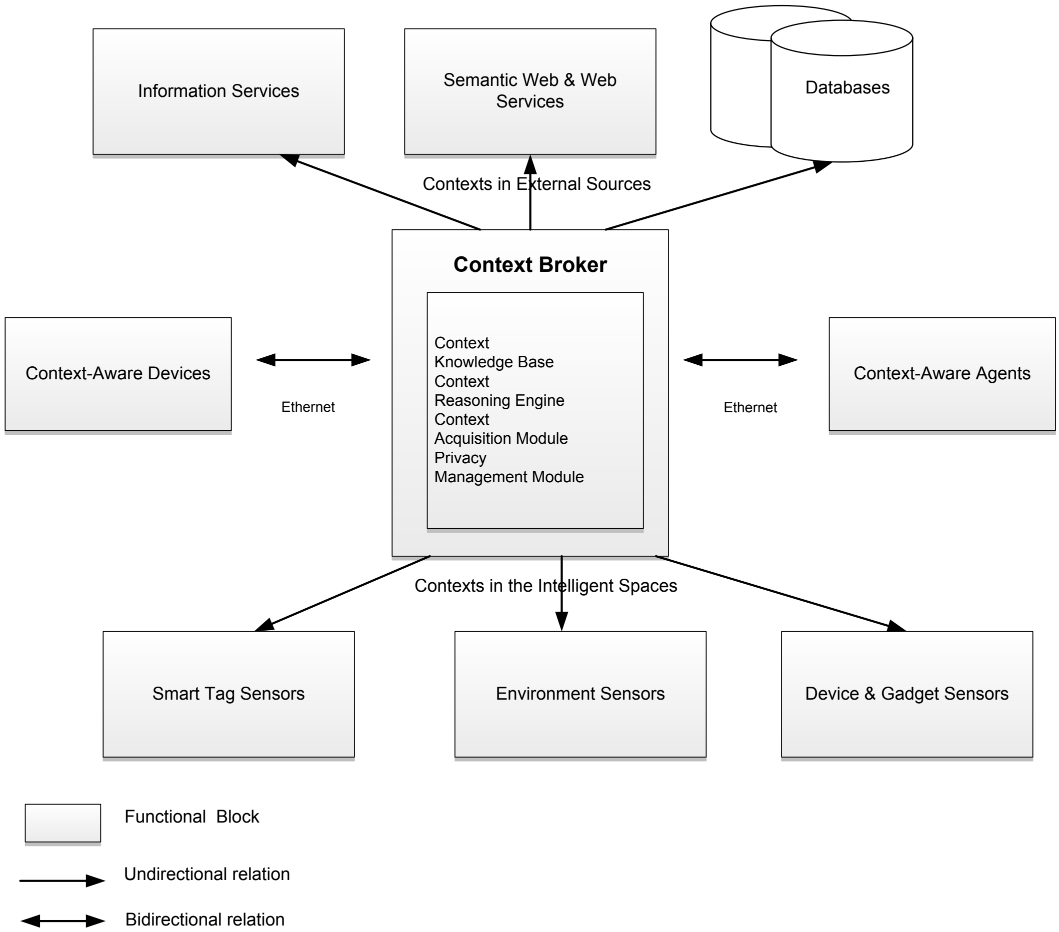

Figure 2.5: CoBra Ontology Using Context Knowledge [HTA03]

CoBra stands for Context Broker Architecture, see fig.2.5 and enables context-awareness in an agent-oriented framework, as the major unit in the model is an intelligent agent (Contex Broker) that is responsible for the management of the shared context of several agents. The major unit consists of a *context knowledge base*, a *context inference engine*, a *context acquisition module* and a *module for privacy management*. It is connected to *context-aware agents* and *context aware devices* via Ethernet [HTA03]. Added to that, contexts in external sources is included, such as *Information Servers, Semantic Web and Web Services and Databases.* Apart from this, contexts in intelligent spaces are sensed by *smart tag sensors, environment sensors and device sensors.*

The context broker model COBRA is at the moment not suitable for the ARS model as there are few devices that are in fact context dependent. Also, the model in ARS architecture including databases is quite simple still. Thus, the context broker ontology is only suitable, if ARS in applied in practical terms and a central context broker architecture is needed for monitoring context in all external devices.

2.3 Computational Models Using Knowledge

Besides using ontologies for context identification the meaning of the situation as such should be covered in the ARS model. To capture the meaning of an object or a situation, it is proposed by domain theory to use ontologies. An ontology is developed according to *domain theory* in a specific domain [AWJ95]. Fig.2.6 represents the development in more detail. Within domain theory, certain mathematical rules are valid. An ontology there describes the perspective, the concepts and rules of a certain domain. Thus, an ontology can be seen as a meta-level to describe domain theories. Domain theory can be applied to an *application domain*, if an *interpretation context* arises. The level of interpretation depends on how accurate the domain theory was formulated. Aside from domain theory other approaches for context identification are used, such as machine learning and pattern recognition methods [Wid97] or by means of classification in computer vision [SS05].

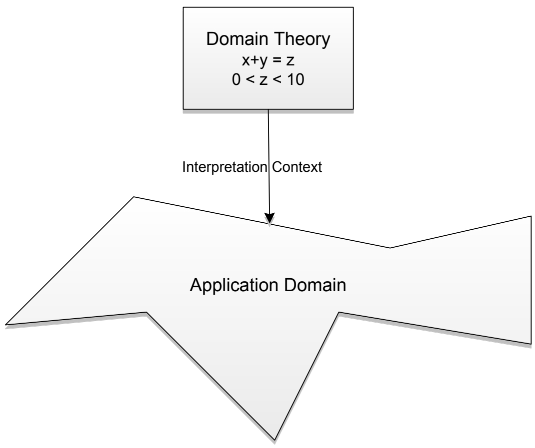

Figure 2.6: One Ontology as a Meta-model of an Application Domain [AWJ95]

Added to this, an ontology can be modularised, if it becomes too complex [AWJ95], as it is seen in fig.2.7. The key point is that the *application ontology* consists of several smaller ontology libraries, such as *ontology1* and *ontology2*, even, if these ontology libraries are conflicting or alternative to each other. The ontology in this model is a viewpoint of the domain theory and adds mapping rules to the domain theory, which is changed then. Both domain theories, the initial one and the rewritten one containing added components, can be applied if there is a *interpretation context* for them. A further possibility is to use an ontology as a meta-ontology for describing domain theory for a certain applicable domain, meaning several levels of abstraction.

Domain theory, as such is used in my thesis as a meta-model for establishing context in an application domain. For the description of the ARS knowledge at a general and a specific level, two different ontologies exist. As seen in Fig.2.7 they are both different ontologies that are imported in a main ontology, namely the whole ontological knowledge which is used in the ARS model. The different domain theories resemble different situations that can occur within an application domain.

One of the first approaches of modelling and defining context has been proposed by Otztürk and Aamodt [OA97] as it can be seen in 2.8. They developed a model for context, based on roles and focus in the field of medical diagnostic problem-solving. They argue that focus is important to

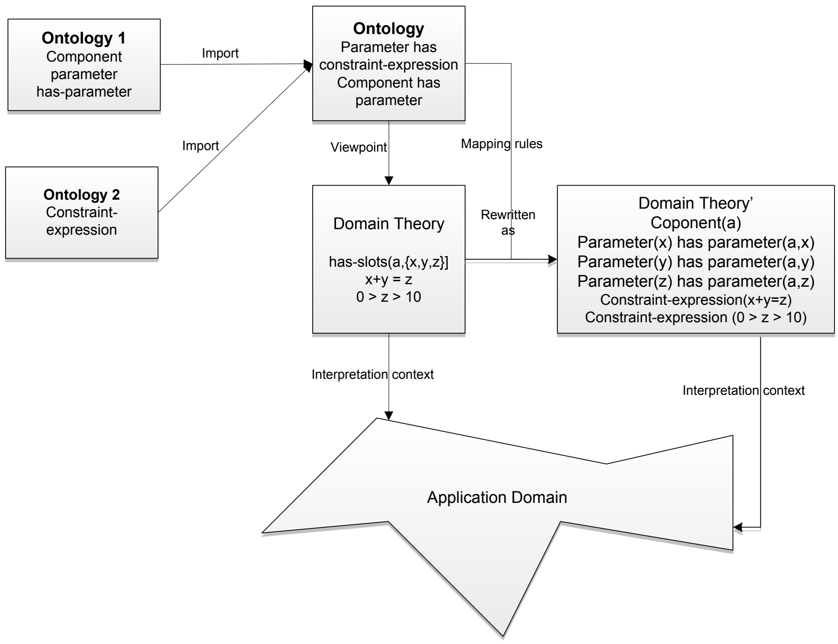

Figure 2.7: One Ontology as a Meta-Model of Another Ontology [AWJ95]

improve efficiency in a search task and investigate on how context is connected to memory usage and action planning. Depending on the memory that is searched for, different types of context access the memory. Further, they separate between internal and external context. Internal context occurs during a learning epoch, whereas external context are grounded facts existing in a certain situation. Internal context is further separated into interactive and independent internal context. For external context, target related and environment-related context is distinguished. Both types of context are separated into interactive or independent context.

Based on these distinctions by a top-down modelling approach, they distinguish between other subgroups in the class of internal and external context. In a more specific ontology, Otztürk and Aamodt modelled an ontology for diagnosis and treatment. The domain-specific ontology contains knowledge of the current task and domain knowledge, as well as a plan case and explanation knowledge base.

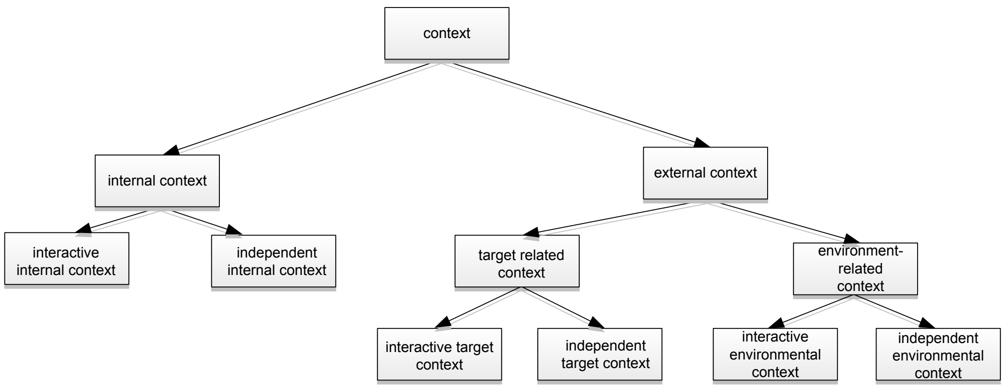

Figure 2.8: Classification of Context Types According to Aamodt [OA97]

The modelling of different types of knowledge - semantic and domain-specific will be explained in the coming chapter. Computational models of semantic knowledge are taken into account. Furthermore, each of the proposed models contributes to the discussion of semantic knowledge or memory.

2.3.1 Computational Models Using Semantic Knowledge

On the one hand *Semantic knowledge* is often described as knowledge about procedures and concepts. According to Clua and Feldgen it is characterised as knowledge of the concepts that underlie the problematic situation, such as what are the concepts of force, mass and acceleration [CF04]. On the other hand, knowledge can be declared as *episodic knowledge*, which is knowledge of facts, episodes and events. Episodic knowledge in humans is acquired throughout life and an agent has to learn episodic knowledge in order to use it. In further works, a distinction is made between semantic and domain-specific knowledge. Domain-specific knowledge resembles the knowledge in a certain domain. For example, reading a book on warriors and gladiators affords domain-specific knowledge. To organise memory, one has to know in advance what a memory trace means. For example, memory refers to processes that acquire, store, retain and retrieve information. Thus, three major processes are involved in memory: encoding, storage and retrieval. According to Solms and Turnball [ST02, p.140] encoding is defined as the process of acquiring new information, storage is retaining the information as such and retrieval is bringing back the information.

Apart from this, the field cognitive sciences deals with the fact that human memory needs to be improved. A lot of empirical research has deals with the fact that human memory is variable and different kinds of memories exist. In the following models are explained that aim to create semantic knowledge by means of methods from computer science. Often semantic knowledge and semantic memory is used in the same context. The term "semantic memory" stands for the memory of meaning [Rog08]. Furthermore, semantic memory stands for all the concepts and knowledge that one has acquired about the external world [E.T92]. For the underlying model in ARS a semantic memory knowledge base existing of domain-specific knowledge is needed and used already. Thus, in the following, models are reviewed that enable the use if semantic knowledge or semantic memory.

- Spreading Activation Models,
- Prototype and Similarity-based Approach,
- Distributed Semantic Models,
- Simple Recurrent Networks, and
- Neuro-cognitive Models.

Spreading Activation Models

One of the first models for spreading activation was the Collins and Quillian [CQ69] model. It is a hierarchical model that consists of three layers. The model contains concepts as nodes, predicates as properties and connections between the nodes. Thus, the structure of concepts and predicates resembles a tree. For example, the concept *Canary* has the predicates *can sing* and *is yellow*. In addition, the model encompasses subclasses and super classes. *Spreading activation* is used for accessing information at a certain hierarchical layer in the model. Then a bottom-up process is triggered and the activation spreads from the bottom layer to the top. Furthermore, the authors propose the hypothesis that if concepts at different levels are linked the tree system is able to provide more economical ways of knowledge storage.

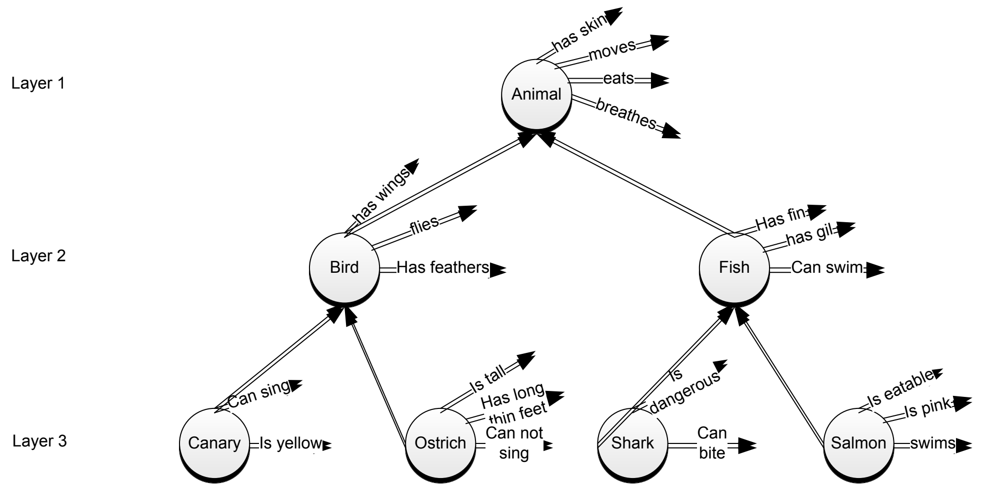

Figure 2.9: Spreading Activation Model of Collins and Quillian [CQ69]

The model of Collins and Quillian has limitations as it is unclear how semantic knowledge is encoded in the network and can be linked especially to the perceptual and motor system. As a consequence, the model was adapted by Collins and Loftus in 1975 in such a way that it starts at the bottom-up of the hierarchy and moves up. The activation spreads along the edges to the higher level nodes [AC75].

In summary, the spreading activation theory has advantages and disadvantages. It is unclear how semantic knowledge is encoded directly in the model of Collins and Quillian. The notion of context is encapsulated in the super nodes. For example, each node presupposes, which context is given. Hence, each node can be seen as a domain and the specific items within a domain are domain-specific context items.

Prototype and Similarity-based Approach

The approach deals with the question of how objects are categorised dependent on the naming and inclusion [Rog08]. However, Wittgenstein at the same time observed that everyday categories and concepts are not formally characterized by necessary and sufficient criteria [Wit53].

Based on his observations on that matter, he proposed that semantic knowledge about properties is stored in a set of common category prototypes. That is why, the approach is referred as prototype or similarity-based approach. The approach differs from the spreading activation approach in several aspects. Spreading activation approaches are organised in a hierarchy of concepts, whereas similarity-based approaches do not presuppose any case of a hierarchy.

To put it more precisely, that means that the knowledge that penguins and fish can swim is stored with the concept animal in spreading activation approaches. For a specific animal, the knowledge is accessed by means of spreading activation. However, it is not clear where the information is anchored in the similarity approach.

Distributed Semantic Models

One of the first semantic models in history has been the Hinton's model that used propositional semantic knowledge in a parallel distributed processing network (PDP). Usually the inputs in the model come directly from the environment and outputs that correspond to the actuators on the other side. Information processing in the model is an activated pattern of activation based on neuronal processing units [Hin81].

The principle in such networks is that the knowledge as such is stored in the weights and through adaption of the weights new information is stored and a new mapping between input and output is achieved. This kind of network is considered a *connectionist network*. The architecture of the model is more closely described in 2.10, consisting of three different layers, an *input layer, an hidden layer and an output layer*. An architecture of this type is also called Item-Relation-Attribute. The connection between the items is the relation.

In the next step, Rumelhart wanted to know, how representations and concepts are accessed in Hinton's model.Consequently, Rumelhart developed a model consisting of a multi-layer network of several nonlinear units that are connected by feed-forward propagation. A pattern is defined by the activation of one unit in the item and relation layer. This activation spreads forward in the network and then modulates the connection weights. The model is trained by means of a back-propagation learning algorithm [RHW86], where the output is compared to the target values and the deviance is calculated then.

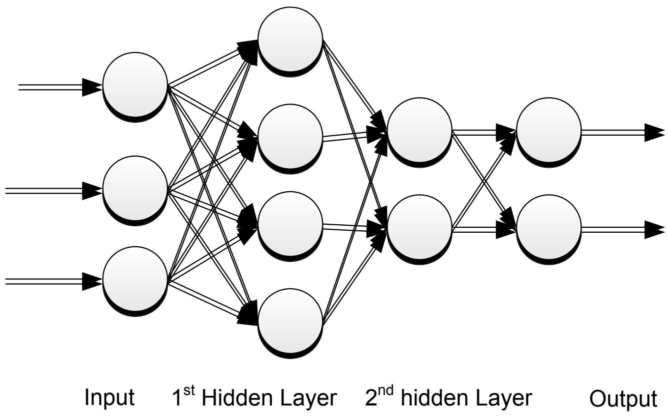

Figure 2.10: Feed Forward Propagation in the Rumelhart and Hinton Model [RHW86]

In summary, the model gives a simple implementation of semantic abilities. The actual semantic memory is represented by internal representations that are discovered by the back-propagation mechanism.

One of the early models in 1991 was a computational model for semantic memory and its impairment that was devised by McCleeland and his colleagues. The model was based upon a parallel distributed architecture. It is a neural network architecture, where semantic memory units are divided into *visual units* and *functional units*. Information is perceived by the *verbal and visual* input system. Furthermore, the peripheral input systems is connected to the functional and visual semantic system. The processing assumptions for the distributed model are the same as in the model of McCleeland and Rumelhart (1985) for the distributed memory model. The model,

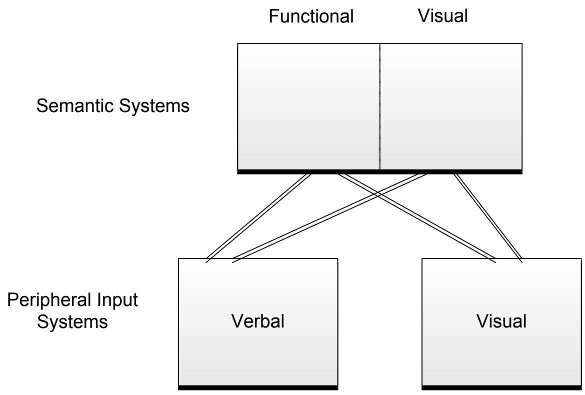

Figure 2.11: Diagram of the Parallel Distributed Architecture [FM91]

however, was used to simulate impairment of semantic working memory [FM91], when the distinct nodes where not active.

In 2004, Rogers and McCleeland proposed a neuropsychological and computational model for semantic memory [RM04].According to Damasio [A.R89] it is suggested that for semantic memory processes, semantic knowledge as such develops from modality-specific, perceptual representations of objects and statements in semantic memory processes. The key aspect is that semantic memory does not need to extract, store and retrieve attributes. Furthermore, abstract semantic representations develop by means of statistical learning mechanisms. That is why the model architecture is implemented with a neural network using the parallel distributed processing (PDP) architecture.

Fig.2.12 describes the architecture consisting of 64 *Visual Features*, units that serve for high-level visual information, whereas verbal units represent the linguistic processes, such as *Names, Perceptual, Functional, Encyclopedic*. Above the visual and verbal units, the *Semantics* units do not receive any environmental and perceptional input. They are also referred to as hidden units. The network is trained and assigned with a non-trainable bias at the beginning. The consequence is that in the absence of input, each unit in the whole neuronal network has a low activation. A training pattern is used for once in an epoch, whereby the model is trained for a timespan of 400 epochs [RM04].

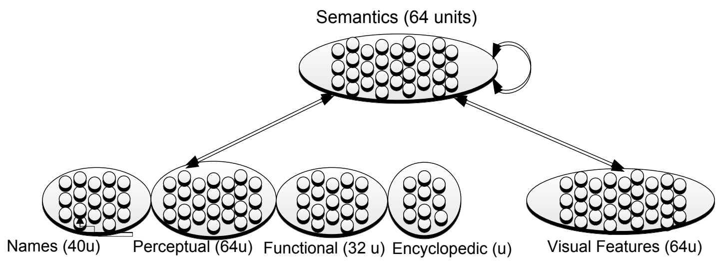

Figure 2.12: McCleeland Model for Simulating Vision and Speech [RM04]

Simple Recurrent Networks

The simple recurrent network (SRN) was developed by Elman [Elm88], who in addition to semantics, was also interested in special aspects of language, such as the segmentation of auditory

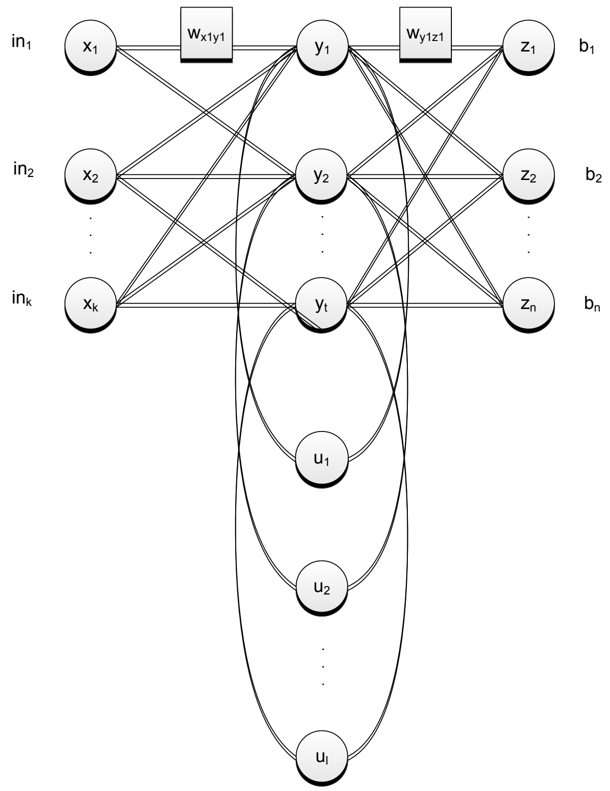

Figure 2.13: Elman Net: Recurrent Simple Network [Elm88]

streams into words or the usage word of order for context in sentences. The Elman net or SRN consists of a feed forward connectionist network. Input layers are also directly activated by the environmental perceptions. The model can be seen in fig.2.13 above. The processing of the activation is transferred forward by *hidden units*, from the *input layer* to the *output layer*. The architecture is considered a recurrent network because of its context layer. This layer is not activated by the input layer, but by hidden units activated in the previous time step. These previous hidden units enable the network to recognise and react towards a temporal structure and thus are seen as actual "memory".

The SRN has the following disadvantage - internal representations acquired by an SRN are context sensitive. Representations stemming from the context layer, contain information on several inputs encountered over time. Thus, an SRN gives insight into how the semantics of the network establish internal representations covering the semantics of a whole event over time, instead of the meaning of one word being matched to one of the samples in the text.

Neuro-cognitive Models

These kinds of models distinguish between semantic deficits due to a module specific impairment of a network, for example if modules are not activated. Researchers investigate the impairment of knowledge in living objects in comparison to an intact knowledge of nonliving objects. The process is known as double dissociation of naming in living and nonliving things. For example, the sensory functional hypothesis was implemented in a recurrent network by Farah and McCleeland [FM91].

The network architecture consists of a three-layer architecture of a visual layer, a semantic layer and a verbal layer. The processing of the input is propagated from the visual to the semantic to the verbal layer. Optionally, the input is processed from the verbal layer to the semantic layer and back to the visual layer. Therefore, units in the semantic layer are considered as mappings between visual and verbal information from the environment.

2.3.2 Context Models using Semantic and Domain-Specific Knowledge

One of the later models is a semantic knowledge model called AGENTOWL. The model has been set up for discrete environments, where the Jena library and the JADE ontology model [FB07] were used for implementation. The aim was to design a multi-agent architecture (MAS), where the agent application needs to be connected with semantic web technologies. The implementation of the agent memory was done using RDF/OWL. For the knowledge model, the CommonKADS was used, which is a structured environment for building knowledge-based systems. The system was implemented using UML, AUML and MAScommonKADS.

The current knowledge model is based on the three parts entailed in CommonKADS:

- The Organisational Model models actors performing tasks and actions,
- The Task Model for representing tasks/activities/processes and actions, and
- The Agent Model based on OWL/RDF

Laclavik [MLH06] and his colleagues developed a semantic knowledge model that covers context in the field of MAS.The main aspect of their model is that it is general and applicable to environments that are observed fully. The model proposed entails *Action, Task, Event, Resource, Actor, Agent and Context*. A task is a context and a task is in a domain, where a domain is described by a context. The relationship between the classes is described in fig.2.14. Moreover, the relation *context - Actor* represents the current context of an actor. Further, events can be stored and thus the current state can be affected by new contextual events.

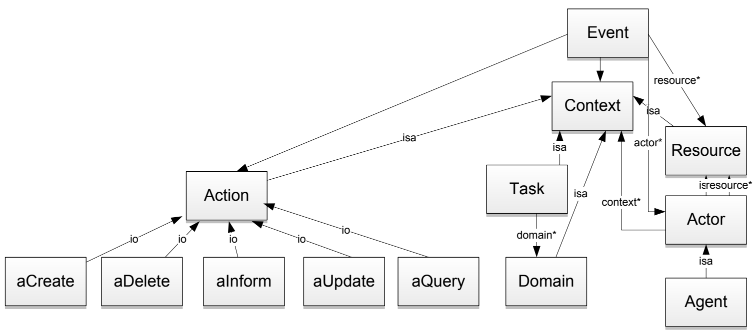

Figure 2.14: Illustration of the AgentOWL Ontology [MLH06]

Another model that uses semantic knowledge is YAGO by Suchanek and his colleagues [HSB+11]. YAGO is described as a general-purpose ontology based on entities (subjects and objects) and relations, which is realised in OWL (Web Ontology Language). However, more interesting in the scope of semantic context is the project YAGO2 [SKW07].

YAGO2 has been developed in a top-down process in order to encapsulate entity-relationship-oriented facts. Their model is based on subject, predicate, object plus time and location. This offers the advantage of a more convenient way of making queries in the YAGO2 knowledge base. In addition, YAGO2 uses selected keywords that are automatically selected from the context. Thus, YAGO2 also has context incorporated. Namely, for every entity in the semantic knowledge base, contextual information is extracted from Wikipedia and anchored as such.

Using general, upper ontologies that have been introduced in this chapter in the section *Context Aware Models using Ontologies* is explored in the review of Baumgartner and Retschitzegger [BR07]. However, in the field of situation awareness (SAW) and context awareness (CAW) also context models using domain-specific ontologies are present.

Relevant work in this field will be presented in the next paragraph. In a semantic network approach, Aamodt and Langseth [Aam94] developed a so-called general domain knowledge in the CREEK project. As an example of general domain knowledge, they used a whole network of concepts and relations represented by nodes and edges. This type of knowledge consists of knowledge developed by humans as well as knowledge created by means of Bayesian Networks. Practically, it means the unification of upper level ontologies of facts and domain-specific knowledge in the form of user experience.

Domain-specific knowledge captures the changing of the problem domain, whereas upper level knowledge is static. Bayesian methods are used to retrieve information, such as similarity calculations. With regard to the ontological structure in the ARS model, the upper level ontology consists of static aspects, as well. However, the domain-specific part recognises the problem domain and triggers an action in the form of speech.

2.3.3 Speech in Cognitive Architectures

As the goal of the thesis is to implement inner and outer speech by means of situational context, the next chapter focuses on the development of language in believe desire (BDI) architectures.

For this reason several architectures SOAR, BDI, ACT-R, and CHREST will be presented in regards of their contribution to language. Besides the two roboter architectures CELL and Kismet will be discussed.

At first, SOAR is a cognitive architecture that is general and flexible for cognitive modelling and building knowledge based agents according to Laird [Lai08]. Besides several advantages SOAR suffers from disadvantages as well. Thus, in the former years new mechanism were added to SOAR, such as other mechanism for learning and non-symbolic processing. The original SOAR architecture consists of a symbolic long-term memory (LTM) and a symbolic short-term memory (STM). At the lowest level SOAR makes processing by matching and firing rules. Regarding language modeling, SOAR has sufficient storage capacity within the memory compounds but the mechanisms for language are not solemly build on matching and firing rules.

The second introduced architecture is called Belief Desire Intention (BDI) architecture and can be seen as a software model to design and program rational agents. The model is based on the fact that intentions are crucial for agent-based practical reasoning according to Bratman [Bra87]. The model consists of four major categories: beliefs, desires, intentions and events. Beliefs are characterized as a beliefeset, whereas goals are desires that are used for active pursuit. Plans consist of sequences of actions and intentions define the deliberative stage of the agent. Last but not least events are triggers for an activity, which can be either internally or externally generated. However, several programming languages can be used for programming a BDI architecture ranging from Erlang [F.D12], dMars [al.99], Jack [PBA99] or AgentTalk [Rao96].

Third architecture presented is the CHREST architecture, which stands for Chunk Hierarchy and Retrieval Structures. According to Gobet and Lane [GP10] the architecture consists of a short-term memory (STM) and a long-term memory (LTM) and input/output units for interaction with the outer world. Latter one is processed by feature extraction mechanisms. Chunking is a problem solving technique that uses perceptual chunks. In addition, a planing mechanism is at work that retrieves chunks that have been acquired for smaller problems. One of the key points in the CHREST architecture is the limitation of cognitive abilities. For example, the capacity of the STM in CHREST is constrained to three chunks. Also the learning rate for transferring chunks from the STM to LTM takes about eight seconds, which is quite limited.

Fourthly, the ACT-R architecture is presented, which is based on the ACT theory from Anderson [AL88]. ACT-R is a theory that varies from the original ACT theory that has been postulated in 1983. The theory claims that a cognitive skill is based on production rules. For example, adding up two numbers acquires a set of production rules. As regards language production the ACT-R model can be used for language analysis, for example when integrating symbolic and probabilistic processing. Practically ACT-R can be used for developing the language analysis model for mapping linguistic representations into a situation model [Bal11].

In addition to that, the robotic architecture KISMET will be introduced. Kismet is a robotic head created at the Massachusets Inst. of Technology, the uses apart from behavioural manifestations of emotions, a sort of prototypical language. The emotions that Kismet is able to represent is anger, frustration, disgust, fear, distress, calm, joy, etc. Kismet's protolanguage module uses different behaviour types that are meant for the satisfaction of gaols and desires. Proto stands for pre-grammatical. Kismet is able to produce verbal output that a newborn can produce at the age of ten to twelfe months. These protoverbal types of language are babbling, grunting and the protolanguage comprising different emotions.

Finally, CELL is using a vision camera and acquires lexical units from a given scenario. In case the vision system detects an object the robot gets colour and shape information from the object. The new vocabulary that is leant does not need to be associated with colour and shape in any case. Thus, CELL can learn words that are not connected with the object. However, the associations then are created based on the maximum of neutral information they have in common [Var02].

2.4 Evaluation of Language Production Models for ARS

Modelling language and speech is popular in computer science and cognitive sciences. Therefore, different approaches for modelling language exist. For example, connectionist models are characterised by the usage of several simple units that dynamically adapt to the environment [BA91, p.30]. This chapter conducts an evaluation of a few computational language models, where the evaluation criteria are the following:

- is general and has concepts,
- is functional or modular in terms of processing,
- is hierarchical (either top-down or bottom-up),
- is extensible to new functions,
- is reusable in another domain,
- is connected to knowledge bases,
- is rule-based, and
- captures semantics.

The criteria are chosen on the basis of employing a speech model in the ARS model.

Roelofs developed a computational model for word production called WEAVER (Word-form Encoding by Activation and VERification) that is based on a retrieval of word forms by spreading activation and the verification of activated information by a production rule. The model generates encoded words by spreading activation through the network. It works in a parallel way together with a knowledge base of linguistic rules or constraints [Roe97]. Words are planned by several functions working in parallel.

Regarding the evaluation criteria, the approach accounts for lexical concepts that are mapped onto an articulatory program. However, a more abstract concept is not mentioned. The model is functionally orientated, as a team of procedures work in parallel on a word. It is a hierarchical model that differentiates between three stages: phonological encoding, morphological encoding and phonetic encoding. Due to parallel processing of functions, it is not easy to extend it to new functions. Initially, the model was set up for the encoding of word forms and thus can hardly be used in another domain. The model includes a knowledge base and rules for the activation pattern. Finally, it can be said that it entails semantics by encoding the word form by activation.

The aphasia model of Dell (see fig.2.15) aims at investigating the errors of aphasic and non-aphasic speakers. Meaning, speakers who have aphasia and speakers who do not have this language impairment. The architecture of the model contains three layers of units: semantic features, words, and phonemes. At the beginning of the lexical access, spread activation starts at the semantic features and spreads according to a linear activation rule. Due to bidirectional connections, all layers of the network are activated. The phonological selection as a second step starts with an activation, as well. If a selected word is active, the phonemes are selected when they are most highly activated [DSM+97].

For the evaluation, the model contains general concepts in the three-layer architecture, such as the entities for semantics, words and phonemes. The model does not deal with functional units, but with an interactive form of spread activation that can spread in two directions. However, it has a hierarchical set-up, as there are three layers. The model can be extensible, although there are no functional units. This is due to the fact that another layer can be modelled for processing. Taking only spreading activation as a function, it is reusable in another domain. The whole model can be integrated in a model matching the hierarchical structure of layers and nodes. To simulate the aphasic and non-aphasic language, the model is connected with a lexicon. However, there are no rules that are used in correspondence with the lexicon. Finally, semantics is entailed in the model due to the semantic features that are properties for selection of the words.

Another model was set up by Dell and his colleagues [GSDG93], the phonological error model to use PDP on phonological encoding. The model is based on a recurrent network, consisting of an input in the form of a context unit, hidden units and an output. Thus, a mapping occurs from a

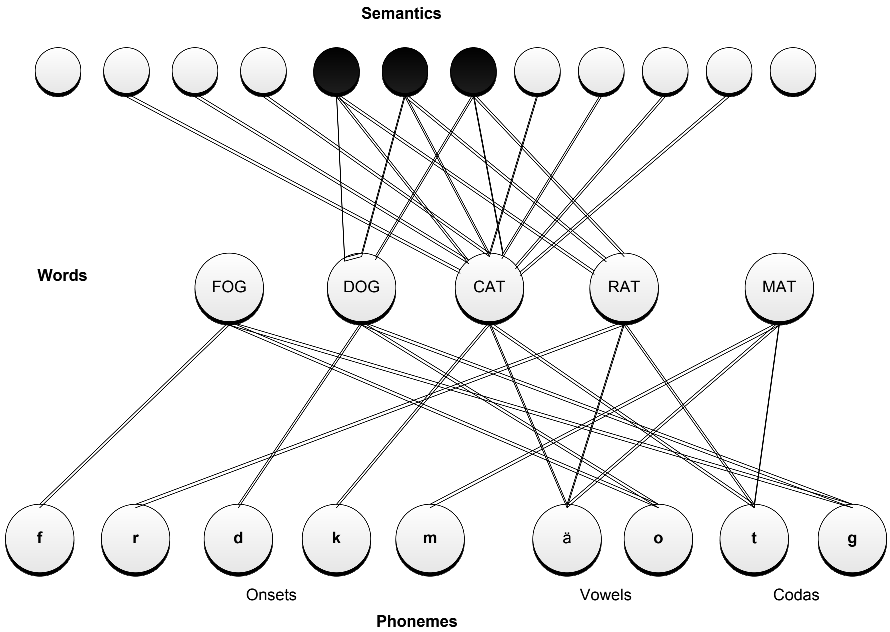

Figure 2.15: Aphasia Model of Dell [DSM+97]

word presentation to a phonological representation. The model is based on spreading activation from the input and context to the hidden units and the output. The context unit keeps track of where the model is located in the sequence of a word, i.e. the change in context triggers the production of the next sequence. The model has been trained on vocabularies by presenting words and adjusting the weights. A disadvantage of the model is that it does not account for speech errors.

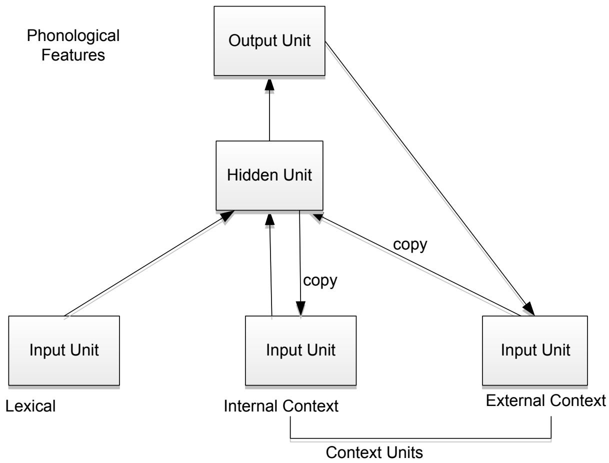

Figure 2.16: Phonological Error Model of Dell and his Colleagues [Del86]

As regards the evaluation, the model uses concepts, such as external and internal context, and lexical input. It is both modular and hierarchical, based on input, hidden and output modules, and several layers. Partly, because of this modularity, it can be extended by other modules. The model can only be applied within the framework of language learning. It is connected to vocabularies for training, which are not treated as a knowledge base and hence no further rules are needed. A spreading activation function is used for mapping external context to the phonological features.

Chang and his colleagues [CGDB97] have developed a connectionist model of language called the structural priming model. As the name says, it has been established to account for the structural priming effect. The main hypothesis of the model is that structural priming is a type of implicit learning, meaning the mechanism for learning is the same as for priming. According to Bock, structural priming refers to the speakers tendency to produce sentences with previously heard syntactic structures. Structural priming in the model is implemented as back propagation.

Context is integrated in the model, as there are special context representations that guide the language production. As can be seen in fig.2.17, the model consists of hidden units and context activation patterns. The input to the model is a message entailing semantic context, and it stays activated throughout sentence production. Apart from structural priming, the model is suitable for connectionist models which derive their structure from learning experience.

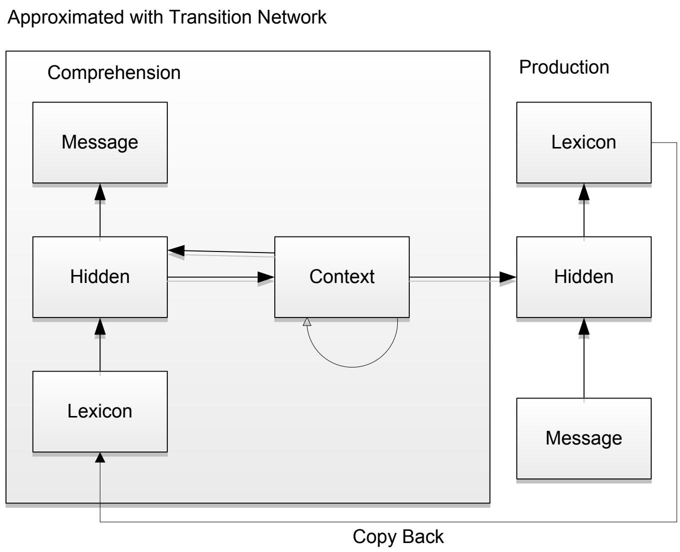

Figure 2.17: Structural Priming Model of Chang [CGDB97]

As regards the selection criteria, the Chang model does not have sufficiently abstract concepts. Though it is functional in terms of processing, the production and comprehension processes are both hierarchical. The model can be extended in terms of other functional modules in the production and comprehension process. It is not easy to reuse it in another domain, because of the aspect that production and comprehension are connected to each other. Also, connectionist networks assume a certain amount of nodes for each layer, so one has to adjust the number of nodes if using it in another domain that language production. The model is connected to a lexicon, which can be seen as a knowledge base. However, there are no further rules employed in the connectionist model. The model entails semantics at the beginning, as the input message consists of a set of local semantic features.

Finally, the speech production model of Levelt is described in fig.2.18. He aims to describe the speech production from intention to speech, where he sees the speaker as a "highly complex information processor." [Lev89, p.1]. The model of Levelt contains three main function modules. The conceptualiser, the formaliser and the articulation unit. First, the message is generated in the conceptualiser, depending on context and situational knowledge. The preverbal message at this point already has semantic character due to the fact that situational and contextual knowledge is integrated. In a second step, the message is sent to the formulator, which is connected with the lexicon and the speech comprehension. The formulator creates the phonetic plan according to grammar and semantics. Third, the inner speech is sent to the articulation unit. An inner loop checks if the inner speech phonetic is valid and sends it to the speech comprehension. Subsequently, the articulator produces overt speech. Another loop sends the overt speech to the speech comprehension system to check for errors.

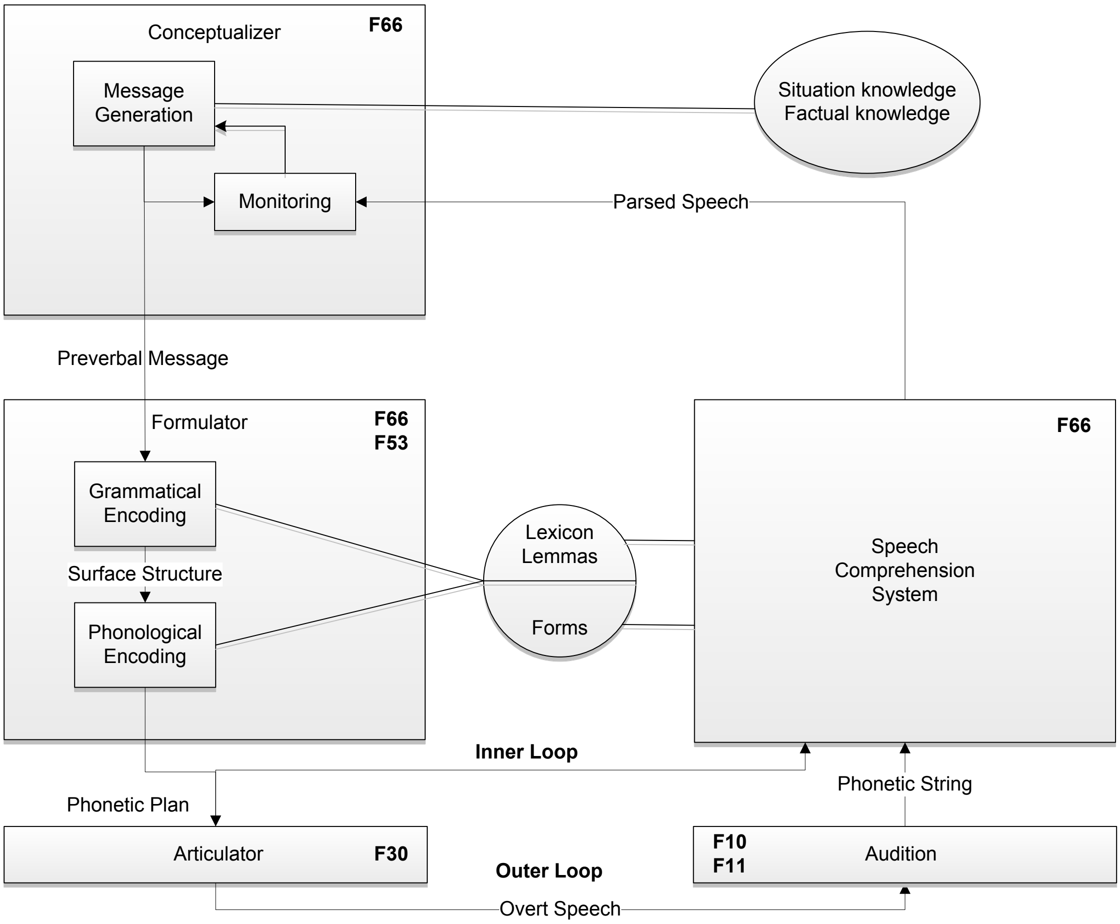

Figure 2.18: Speech Production Model of Levelt [Lev89]

The selection criteria were derived from a description of the ARS model. They are based on several characteristic features of the ARS model such as abstraction ability of the model, functionality, extension, hierarchy, reusability of functions, access to knowledge base, rules and inferences and semantics. Semantics addresses the ability of a model to describe the meaning by using a structure, such as a certain language would be.

Thus, the models of Waever, Dell, Phonological Error Model and Chang were evaluated on this basis. The Dell model showed several overlappings as well, because it has a general architecture.

It is built up in several hierarchies of layers and is rather flexible regarding extension. Also several functions of the Dell model can be reused due to the modular architecture. Apart from this, it is based on a knowledge base and describes semantic relations. The Levelt model consists of a general architecture due to its three layered architecture. It is based on a top-down approach of modelling and the layers are connected in their own hierarchy. Meaning there is a bottom layer and a top layer. The model is extensible to new functions or knowledge bases and it can be easily applied and reused in another domain because of its flexible character. Last but no least it is connected to a knowledge base. During the formalizing step several rules come into play and thus the model is also rule based. Semantics is integrated due to the fact the produced speech captures the meaning.

In summary the evaluation of this chapter shows that the model of Levelt is most suitable for the ARS model, because it resembles a meta-model for speech due to its modular character. The model of Levelt can also be applied to any arbitrary symbols, which offers a great flexibility if it is adopted. Furthermore, the model of Levelt aims at transforming intention to language, which is an approach to that can be of use in the ARS model. Integrating Levelt's model means that thing presentations have to be processed, which are then transformed into the belonging word presentations.

In conclusion, the table 6.2 below compares the evaluation criteria for all aforementioned models.

Criteria	Weaver Model	Dell Model	PE Model	Chang Model	Levelt Model
Abstract	x	x	x		x
Functional	x		x	x	x
Hierarchical	x	x	x	x	x
Extendable		x	x	x	x
Reusable		x			x
Ontology	x	x	x	x	x
Rule-Based	x				x
Semantics	x	x		x	x

Table 2.1: Evaluation of Speech Production Models

2.5 Applied Fields Using Situational Context

The field of smart homes already benefits from taking into account semantic context, ontological methods and agent architecture for enabling a context-awareness in their field. Building automation using smart home scenarios have gained popularity in recent years. For example, solutions have been developed to assist inhabitants in saving energy, but at the same time to provide them with comfort. The challenge in building automation is therefore to bridge the gap between saving energy and optimising the comfort of people living there. An approach in this direction is the usage of knowledge bases and multi-agent systems for managing the great amount of data in such a scenario. The ThinkHome approach using ontological architecture and MAS is used because of the rather complex environment consisting of home automation, knowledge representation and context awareness being unified in one model. The multi-agent architecture consists of a User Agent maintaining the environmental conditions for its inhabitant. It is supported by a Context inference agent that is responsible for generating a user context in time and space. The contrast conflict agent solves potential conflict situations between different user goals regarding energy needs. Context is generated in the ThinkHome by means of using ontologies and sensor environments. However, the work creates a perfect synergy by applying semantic context in residential homes and making the building more user-centred [RKIK11].

Recently, other approaches have also included context awareness in intelligent home environments. Retkowitz and his colleagues use an approach of generating a domain-specific context in a heterogeneous environment of services [RP08]. He uses structural and semantic adaption in order to establish an interoperatibility of services that is not possible nowadays due to standards in the field. Chen et. al. [CPC+04] developed the EasyMeeting system based on the Context Broker Architecture (CoBra) for the purpose of capturing the context during meetings. The CoBra enables the role of context for all agents and services. The architecture consists of a context knowledge base, a reasoning engine, an acquisition module and a policy management module for inferences if agents can access a particular part. The selected CoBra architecture has been chosen, as it supports the integration of context and context reasoning. Benta and his colleagues [BHCC09] proposed an ontology for the establishment of context to support behavioural changes according to the usersśfeedback. Consequently, they work on issues related to context ontologies supporting agent communication, sensor integration and affective sensing. The Smart home environment was created using the JADE architecture and using Protege as a tool. The JADE agent platform is central for and supports context, reasoning, actions and sensors at a bidirectional interface.

Another example for the use of situational context comes from automotive industries. For the detection of a situation on the street a module is needed that recognizes this context. Context is interpreted after several sensor informations come togethers in the car and are preprocessed. The more sensor are used the broader a situational context can be. Based on the preaggregated and preprocessed sensor information the situation context is calculated. In another step, yet another modules decides to make a reaction based on the situation context that has been interpreted.

3 Concept

"The degree of one's emotions varies inversely with one's knowledge of the facts - the less you know the hotter you get." Bertrand Russell

For clarification reasons, it will be explained first why the ARS model needs *use cases*. Afterwards the structure of statements in thoughts and statements in speech will be described on the basis of scenes. At the end of each section, a diagram addresses the sequence of each process. This chapter describes existing data structures and new ones that have been introduced to the ARS model. In addition, the role of instances and the usage of spread activation is mentioned. Last but not least, the organisation of the ontology is explained based on the example of the general upper ontology and the domain-specific ontology.

3.1 Development of Use Cases

A use case serves the purpose of testing content-related psychoanalytical requirements in the ARS model. Apart from model-based testing in software engineering, the use case is defined by psychoanalysts [DBM+13, p.21]. Furthermore, the methodology where use cases are defined comes from the psychological field. In the ARS model the psychoanalytical use case represents content-related requirements for the development of a bionic system according to the nervous system. For that reason, two use cases as such have been designed and reviewed by our psychoanalysts in the ARS team.

Three main use cases were defined in an overview paper [DBM+13, p.21]. Everything starts from an initial start condition for the agents in the first use case. The separation into three use cases has been brought up due to the complexity of the mental apparatus as not all requirements can be tested in one use case. Only after the results of a use case have been validated by psychoanalysts after the implementation, a further itertation step can be done. This way greater complexity can be achieved. The first use case has been simplified on the object and interaction basis, meaning fewer objects are used and interactions can be established in order to represent basic processes in the mental model. However, in this work only two use cases that are relevant for speech and communication are introduced here from the overview paper.

Use Case 2 - Adam looking for Food

The initial situation in the first example is that Adam is hungry. The use case (UC) one contains four objects:

- the ARSIN Adam,
- the ARSIN Bodo, and
- the food source,
- the house.

The food source has been defined as Wiener Schnitzel, which is a typical austrian dish. At the beginning of the situation, neither the food nor Bodo are within Adam's focus of attention.

The scene is perceived from Adam's perspective and, in addition, attention is focused on the feeding drive, the drive of self-preservation and reducing drive tension. Generation of pleasure and prevention of unpleasure are the underlying factors.

The scene is set up as following. Adam is hungry. Consequently, he has a low blood sugar level, which produces homeostatic tensions in his body. However, the homeostatic tensions increase while Adam is not eating. He is not aware of a food source that is within his reach and he is not very used to the rise in drive tension.

Adam has an impulsive personality, since he was nursed as a child instantly every time he was hungry. Thus, he had no option of learning to delay his food drive. The increase in drive tension makes him unconsciously aggressive, meaning the aggressive component of the drive tension is higher than the libidinal. Based on his education and impulsiveness, Adam has a quite strong superego. This instance is now activated, after the drive tensions become too high. tict arises because the superego says: you are not allowed to eat and the claim of the ego is: I am hungry. I need food right now. The ego mediates by transforming the aggression into fear for the super-ego and by reducing the hunger and finding realistic ways to search for food.

Thus, the aggressive component is not allowed and the mediation by the ego takes place via the defence mechanism fear. Drive energy is reduced from the drive track and is directed towards perception and thinking processes. The original quota of affects of the drives is split up. Due to the inner perception, hunger and fear become conscious, meaning content and affects are connected to word representations and received by means of desexualized drive energy. Further, perception is focused on eatable objects. The psyche invents a so-called *Probehandeln* , in which the actions for finding food are reviewed. A decision has to be made in the end in order to control the motility.

Adam moves near the food source that is a Wiener Schnitzel and is now aware of it. Conscious thought processes start as to whether it can satisfy the need for food or whether one can get stomachache from eating it. Finally, Adam moves towards the Wiener Schnitzel. However, now he becomes aware of Bodo in his visual field. Unconsciously, the memories of Bodo and relevant chains of memory, which range from Bodo to his older brother, are activated. At this point, the unpleasant memories of Bodo and his brother are associated, for instance that the brother was a troublemaker and he had to share everything with him. As a consequence, he was afraid, but he was also jealous of him. The former quarrelling over the Wiener Schnitzel triggers additional aggression.

Adam's strong superego is activated by means of this aggression. Consequently, fantasies of fighting and being beaten up are activated. The ego transforms the aggression into anger by using defence mechanisms and the anger into fear and feelings of guilt. The confrontation awakes memories of former fights with his brother, which are associated with drive contents from the sexual track. Former confrontations between the brothers were solved by Adam giving in.

Fear and feelings of guilt become conscious through word representations, for example. It does not mean that all associations become conscious in that context; only Adam's negative feelings become conscious. Bodo is recognised by means of the word representation of his name *Bodo*.This is the point where recognition of context and the selection of a word take place. The Adam's thought process kicks in and elaborates in a conscious way the connection between fear, feelings of guilt and Bodo in a hunger context.

From this hunger context, the following alternatives are possible:

- Adam takes the Wiener Schnitzel and shares it with the other agent,
- He takes the food and eats it on his own,
- Adam gives the Wiener Schnitzel to the other agent,
- Adam attacks and bites the other agent and eats the food eventually on his own,
- Adam attacks the other agent by means of words and fetches the food for himself,
- Adam stands still and thinks about what do do,
- He convinces the other agent to fetch the food and share it with him, and
- He convinces the other agent to fetch the food and not share it with him.

Due to the fact that use UC one is not suitable for language and communciation scenarios the second UC has been developed. The second UC has integrated a language component, where firstly inner speech and outer speech are mentioned in a formal UC. However, this UC was an intermediate version and thus the third use case has been developed which can be found as well in the technical review [DBM+13]. The third UC will be introduced in the next paragraph.

Use Case 3 - Bodo - Adam - Bella and a Food Source

In total, UC three subsumes three different agents: Adam, Bodo and Bella. In addition, there is a schnitzel. The scene takes place in the house. All agents have different personalities and by means of their memories they know each other.

The following scenes can take place:

- Bodo alone with the food,
- Bodo is away, Adam and food,
- Bodo returns, Adam and food,
- Bodo, Adam, and Bella, and
- Bodo and Adam away, Bella and food.

Bodo is alone with the food source. The drive tension is high in expectation of meeting Bella and having sex. This generates fantasies that are passed through defence by means of daydreaming. Feelings of pleasure are present, but what about feelings of love? He remembers occasions when he met Bella in the past. The hunger is strong and affects are reduced and navigated partly towards the drive track. The other part of the tension is used for getting attention and planning the event of meeting with Bella. A conflict is that he should not eat the food source because it is supposed to be for Bella, as well.

As regards Adam's personality the following shall be emphasized. For profession, Adam is a writer who is not very successful, because of his impulsive personality he can not stay focused on one thing. In addition, he has self-doubts and can be depressive at times. Adam has known Bella since her childhood. They have been together as a couple. Adam has been left by Bella, because of his personality and for Bella to achieve higher goals.

The scene starts. Adam goes towards the house and thinks about writing in this house. Unconsciously, he smells the food source.

As regards Bodo's personality, he is a successful politician and popular in the media. Although he is very young, he has just been promoted today. He is not in a short-term relationship, but for a short time he has been secretly meeting with Bella. However, Bodo could even imagine reveal their relationship to the public, as they make a nice couple. Bodo is analytical, showing less emotion without going into depth with feelings Nevertheless, he has a crush on Bella, as she reminds him of himself. Bodo does not like people who outperform him, as he has always been number one.

In the scene introduced, he has ordered a Wiener Schnitzel in order to celebrate his promotion, the food has been delivered and is waiting in his house for his young girlfriend. Bodo has not eaten a lot during the day, but he has been celebrating and drinking alcohol for some time. In the scene, he looks for Bella by looking out of the house.

Bodo is alone with the food source. The drive tension is high in expectation of meeting Bella and having sex. This generates fantasies that are passed through defence by means of daydreaming. Feelings of pleasure are present, but what about feelings of love? He remembers occasions when he met Bella in the past. The hunger is strong and affects are reduced and navigated partly towards the drive track. The other part of the tension is used for getting attention and planning the event of meeting with Bella. A conflict is that he should not eat the food source because it is supposed to be for Bella, as well.

Eventually, he thinks of taking a bite inconspicuously and then turning the food so that the missing piece cannot be seen. Finally, he places the food source near the door and leaves the house in order to see Bella. In the meantime, Adam enters the house (This has already been described in the first use case). Bodo's perspective is added. Bodo has been outside and is surprised when he sees Adam at the food source. He does not know Adam, but he fears that Adam recognises him from media and his relationship with Bella has now been revealed. Bodo is blushing, but now his emotions change to anger, as he reminds himself that he only had one hour with Bella and now it is too late. He does not want to be interrupted and starts thinking about how he can get rid of the intruder. In a dialogue, Bodo asks if Adam has been invited. However, Adam lies and tells him that he has been invited in order to get a piece of the food source. The situation now becomes unpleasant for both sides. Bodo stops being sexually aroused and gets angry and disappointed.

Bella now comes into the scene to meet Bodo. They had planned to meet for a date. Bella is slim and pays a lot of attention for her looks. However, as a child she was heavier and she changed a lot in the last stage of puberty. She has learned to repress her hunger feeling by this time. She is proud of her beauty, as she is very narcissistic and focused on her body. She desires the acknowledgement of her beauty from men. She likes the fitness component and realises that she arouses men. She likes Bodo, because he is wealthy, has influence and admires her. At the moment, they meet secretly, but she is convinced she will be soon be his number one once she has seduced him.

Bella grew up with Adam and he annoys her when they meet. Bella practically despises him; unconsciously she admires him for his impulsiveness. Bella is has a feeling of being in an empty body and she does not have direct access to her body and her feelings. She is not aware of hunger and emotions, but she recognises sexual his expectations. In her fantasies, she visualises herself as rich and powerful if close to Bodo. Now she enters the house and is surprised. She sees Bodo, the food source and Adam. Bodo welcomes Bella as his secretary and asks about the other guests. He asks Bella if she knows Adam, as he has announced that he has been invited. Adam takes advantage of the situation and knows the game.

Impulsively, he starts attacking Bella verbally. However, Bella stays calm and asks him to explain what he means. Moreover, she wants to know from Adam what he is doing. In his psyche, Bodo is feeling frustrated, hungry and jealous. He fears that his affair with Bella will come out and so he wants to prevent Adam finding out the truth. There are three alternative actions now. One is that he attacks Adam verbally and wants to fight about it. Another alternative is that Bodo stays calm and rationally forces Adam to go. Another variant is that Bodo goes himself because he is already too late.

Regarding the alternative of Bodo and Adam fighting, Bella is sad and starts eating the Wiener Schnitzel on her own. Concerning the alternative where Adam is forced to go, Bodo and Bella stay, have intercourse and eat the food source in the meantime. With regard to the last alternative, Bodo leaves the scene, and Adam and Bella stay, have intercourse and eat the food source.

3.2 Recognition of Situational Context in Scenarios

Within the first use case, it is essential to capture the semantics of an object in order to establish a situational context. Situational context in this work is defined by specific object-subject configurations from one agent's perspective. Situational context is always viewed from a one-person perspective, as this work does not aim to simulate group behaviour. Fig.3.1 shows the scene the situational context is based on.

In the example below two agents called Arsinis are part of the scene. The agent Adam is in the house and observes the Wiener Schnitzel and the agent Bodo, who is nearby. The situational context can be manifold, but, in connection with the ontology and due to the memories of the agent, Adam knows that Bodo owns the house in this scene. According to his associated memories, Adam recalls that Bodo is a successful person who is not keen on meeting him. Consequently, in the presence of Bodo, he will not eat the Wiener Schnitzel, because the situational context has awoken bad memories. The example at hand describes in a vivid manner what situational context is about. Based on this situational context, the basics for developing thought and speech statements are developed. The following chapter explains the concept of how situations with a certain semantic meaning invoke agents to act by speaking or thinking.

The situational context in this scenario is determined by means of the objects present in the scenario. This situational context is the trigger for a speech or thought statement. The output of the established situational context is an action, either a speech action or a thought action. Within the next cycle the uttered speech action is again taken into account for the situational context recognition. This way the, the speech of an agent is always interpreted in a context and not just

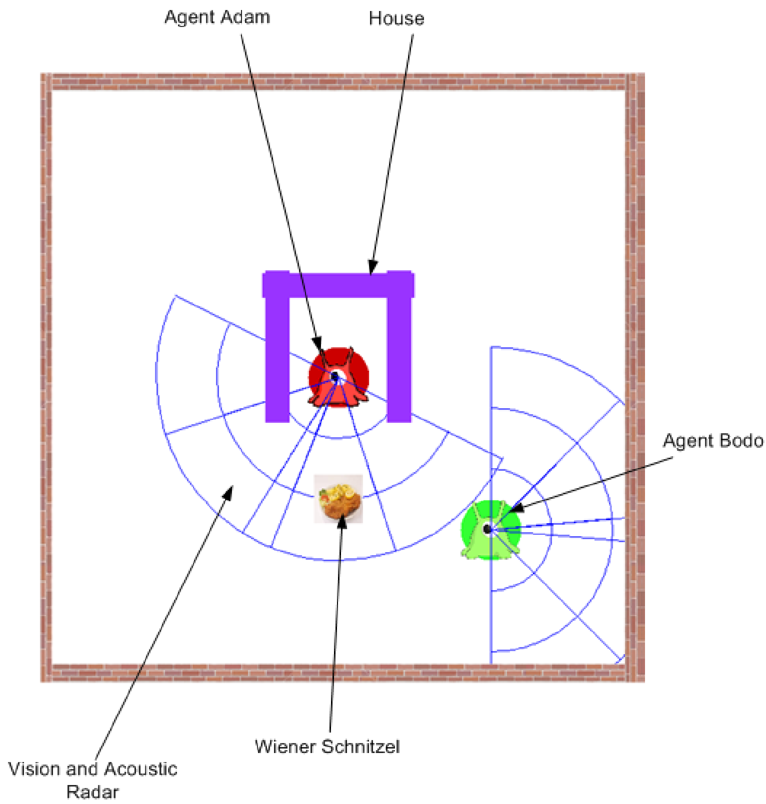

Figure 3.1: Scenario in the Simulator

alone standing. What is needed to develop situational context at first? Basically, the input is twofold, consisting of the actual perception of the environment and the specific associations from the agent's memory. The perception of a situation includes what is seen within one agent's vision radar. This perception is called a configuration within this work. It is translated to an abstract concept and compared to the memory of stored configurations. The stored memory configuration contains all possible context situations described by contextual entities.

When there is a match between the actual perception configuration and a specific memory of an agent, it can be assumed that the current situation has been experienced in the past. If this is the case, then a connection is established between the current situation and the specific memory in the ontology. Then the agent retrieves the statement that is appropriate to say. When there is no full match between a perception and a specific memory, the situation is new. Still, the agent needs to be able to produce speech in new scenes as well. An association mechanism is looking for matches in a new situation until an association is found. Associations are connections between objects and can be retrieved by means of spread activation. So, in case a situation is not known, a known speech statement can be transferred to a new situation. The agent will react upon the association to the object. Associations to object are also stored in the knowledge base.

Another reasoning mechanism is used, if objects in the simulated world are not known by the agent. For instance, take the example that Adam observes the house and the Wiener Schnitzel. Given that Adam knows that the house belongs to Bodo, he follows that the food belongs to Bodo, because the food is in his house. This can be modelled using the ontological reasoning mechanism of Protégé [HOM04]. The class *own house* contains, food and all other objects in the house. If an object is detected and it is in the *own house* class it is presumed that the object belongs to the person the house belongs to.

It is important to know that in the framework of this work situation recognition does not imply learning new context configurations. Learning is not integrated, because it would mean a continuous change of memories in the ontology (knowledge base). Furthermore, it is not possible, because new perceptions that are not in the ontology yet, are not stored dynamically if there is no match. In addition, learning is not integrated because it's a too cost intensive process.

To understand the whole process, the following flow chart fig.3.2 illustrates steps that have to be carried out to identify situational context and develop inner speech. Firstly, input comes in via acoustic and visual perception e.g. outer speech and is processed. The bold numbers in each functional module indicate the module where the function is carried out. The modules are explained further in chapter *Design Methodology and Implementation*. Each speech statement is received as a sound wave and is transformed to a symbol in the neurosymbolic network layer. The next step is that symbol is recognised, and as a thing presentation, it is passed on in the primary process. Finally it is passed on to the secondary process as a word presentation (WP). All modules shown in the chart are part of module F66 in the secondary process. The interfaces within the module F66 are stages to explain the different processing steps.

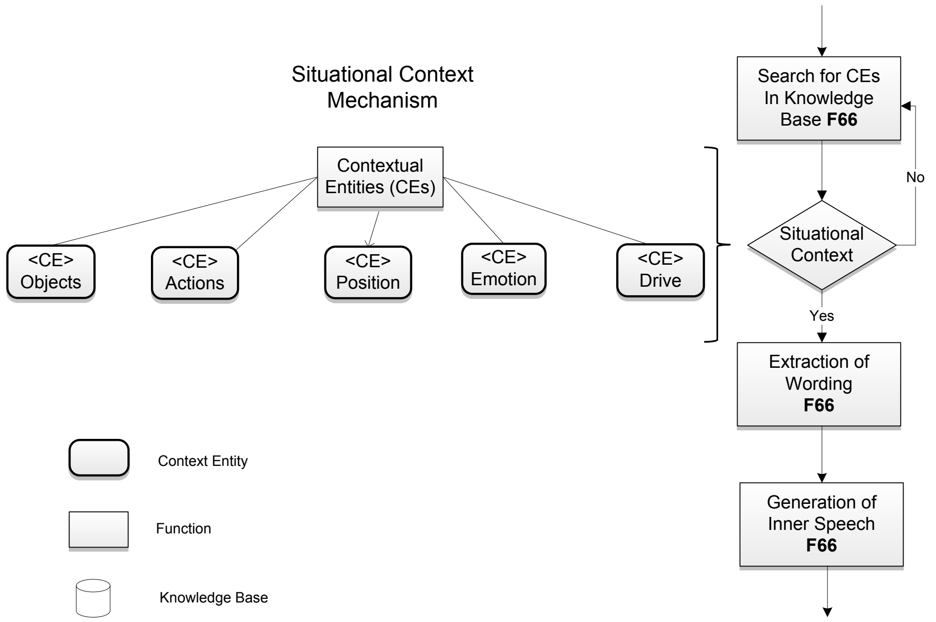

Figure 3.2: Process Situational Context

The situational context is generated by five contextual entities (CEs). A *contextual entity (CE)* is a function that manipulates data itself. The expression entity comes from data modelling in computer science and refers to an entity in a knowledge base. Thus, each CE has properties , such as “objects”, “persons”,“own actions”, “own relative position”, “own emotion”, and the “own drive goal”. An example, for CE Action is *Eat*.

Each TPM can be split up into four contextual items and an additional one, which is the CE:Drive. The CEs from the perceptions are compared to a stored configuration of CEs in the database. If they match with a stored configuration context is identified, meaning the configuration is known. If the CEs do not match fully with the stored configuration in the database, another search for CEs is done until a situational context is found. Also, an abstraction is done in order to make an association with any known object in the situation and react on the object. Meaning in a new situation agents react upon associated objects and only later if they find something they know, they produce speech in a context. However, the search for the situational context goes on in the next cycle.

As a result the output is an inner speech statement that is based on the situational context. The inner speech statement is connected to a situation. For example, seeing the Schnitzel the thought statement is Schnitzel. The stored configuration is the memory of the agent and for this option the following information is stored: the precondition as an action e.g. speech, the emotion, the entities, the distance and the consequence as another action e.g. inner speech or outer speech, and the drive. This information is stored with a *wording* of how to pronounce a word. The drive instance in addition is needed for the actual outer speech. The drive decides, whether something that is inner speech is produced as outer speech. Meaning, depending on the drive compound the agent decides whether to say something and how to say it.

In summary it can be said the situational context sets the content of speech and thought and the drive is the motivation for speech.

3.3 Generation of Statements in Inner Speech

In the first example, speech is not part of the use case. However, thought or inner speech is required in the agents' world as well as a prequisite for outer speech. Inner speech or thought is a word presentation that is associated mainly with a visual and/or acoustic pattern. The difference is that thought is not yet articulated. Also there is no structure due to fact that thought statements are located in in the primary process.

Thoughts have the property that they are produced silently without people paying much attention to them. Although thought processes according to Levelt are monitored, meaning before we utter something we have in mind what we want to say. The statement is prepared already one step before and this is what is considered as inner speech.

Asking the other agent implies that he wants to share the Wiener Schnitzel. When both agents share the food in a next action step, the context is in this situation is shared between the two agents. This kind of common context is also called *common ground*. It means two people, who share their concepts have to know what the other person means by using a word or concept. They need to exchange their concepts or know them beforehand. If two people do not know the concept of the other they can not develop common ground as they do not have an understanding for what the other person means.

Speech statements are developed by the speech production model of Levelt. The psychologist has designed a speech production model that was explained in depth in Chapter *State of the Art*. It fits quite well to the ARS model, as it represents inner and outer speech resembling thoughts and speech as such. The development of speech in this work does not deal with the development of language. Speech statements are simply one-word statements that are triggered by a specific situational context. The structure of a speech statement consists of one speech unit that is based on an inner speech or outer speech statement.

In this work inner speech is defined in the following and is based on the original psychoanalytical expression. *Inner speech is a preconcsious or conscious usage of word- and thing presentations that are formed in the form of sentences (in this work one to three word sentences). These setences can be formed by means of semantic relations. Inner speech takes place in the preconcsious planing and in the form of mumbling sentences or words.*

Outer speech is defined in the work by the following statement. *Outer speech is motorically articulated und articulated speech as an action. It is conscious, but can be unconcsious as well in the case of speech errors. Important is that inner speech predominates outer speech.*

The situation frame contains the contextual entities (CEs). The incoming TPs are matched to CEs. e.g. the *CE: Schnitzel* is described by the concept schnitzel in the memory, whereby the concept is presented by means of the attributes associated to a schnitzel. For example, the schnitzel is described by means of the color and the taste and the shape. A situation consists of five CEs. The contextual frame can be extended by any CE that contributes to situational context. Meaning, a context frame is set in the beginning and thus defines the frame of a situational context. It is possible to extend this frame and have a broader perception of a situation then.

Along with the identification of the CEs, the context is set, meaning CEs are required for recognising situational context, whereas concepts are the fundaments of speech and general or abstract thought. To access domain-specific CEs, a perceptional input is needed. After perceptional input enters, context is retrieved from the WPM and the five main CEs along with the drive content. Followingly, a statement in the form of outer speech can be uttered appropriate to the situation.

The differentiation between inner speech and outer speech is whether a statement contributes to the pleasure principle in a certain situation. If a statement does not enable an increase to an agent's individual pleasure level then it will not be produced. Thus, two ways of processing are differentiated: inner speech and outer speech. Both depend on the situational context, but inner speech is a prequisite for outer speech. For outer speech it is decided whether to speak or not according to the pleasure level increase. The content of the speech statement is determined by the drive component. For example, ff the drive component is libidnous than the agents' response is positive as well.

The next fig.3.3 shows the development of inner speech based on situational context. To generate inner speech, situational context needs to be identified. It is exactly the circled processes in the middle of the flow chart. Situational Context is established when all five CEs are known and TPs are retrieved from them. Only one TP is selected finally as an inner speech statement. Thus, a major assumption is that situational context predominates inner speech.

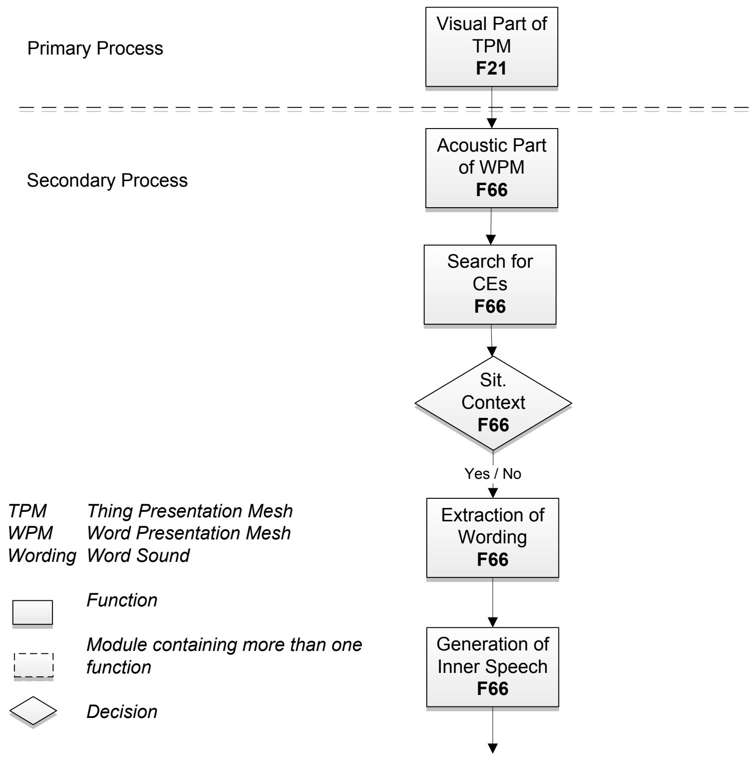

Figure 3.3: From Situational Context to Inner Speech

An algorithmic procedure compares all TPs/CEs to the known concepts and the memory traces in memory to determine whether a certain situational context has been stored there. In the first step, the algorithm works like a greedy algorithm. [1] The concept is compared to the knowledge base, because every item has to be known before identification of the context. Then the CE is mapped directly with the information from the concept known. For example the *concept: Wiener Schnitzel* is mapped to *Entiy: Wiener Schnitzel.* After all five contextual entities are known one can determine the situational context and produce inner speech. If they are not known we can not remember a situation until the next cycle eventuelly provides a known constitution.

3.4 Generation of Statements in Outer Speech

The second process is the outer speech process, which is explained in the next subchapter. Furthermore, the development of outer speech does require access to an ontology. Outer Speech in this work solemnly consists of one word staments, such as: “Hungry?”, “Help!”, “Attack” due to the fact that grammar in speech is not treated within the work. After the produced outer speech, the second agent in the situation does reply in a context dependent way. He takes into account the five CEs and in addition the uttered speech statement and produces a statement when he recognises the situation.

[1] A greedy algorithm terminates always and has a linear growing runtime. It searches for a local optima in order to find a global optima.

As already described in the previous chapter the generation of outer and inner speech needs domain-specific information that is retrieved from the ontology. For an observer situational context is always established via outer speech in correspondance with the whole situation. This chapter discusses the fact how outer speech is in accordance with the situational context.

The process, in fig.3.4 outer speech is based on the situation context, meaning speech can only be uttered in the framework of a situational context. After this context has been identified a speech statement is retrieved from the database, which is a wording. The wording in the form of a WP is produced as an inner speech if the pleasure level rises. Finally, the drive component is responsible for the content that is selected. Outer speech is generated if the pleasure level increases rather than decreases.

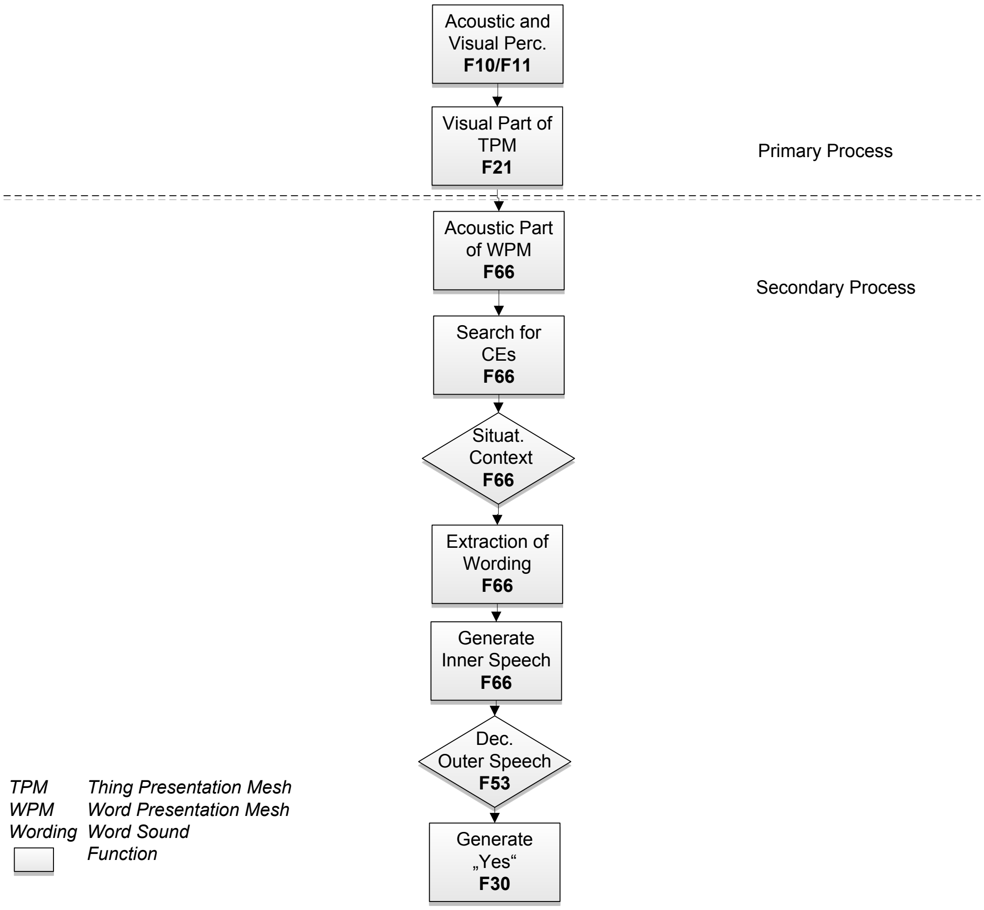

Figure 3.4: Process of Outer Speech

Speech comprehension of context is accomplished if the listening agents understands the meaning of a situation. For example, if an agent produces an utterance and the opposite agent reacts adequatly within the recognised context, then semantic assignment between context and context entities has been established successfully. The agent may react in different ways as long as it is within the situational context scenario.

Thus, semantic assignment comes into place after the situational context and the speech act has been established. Prerequisites for semantic assignment are the recognition of situational context and the ability to utter a speech fragment. Semantic assignment is needed further for the development of communciation in between agents and for the development of language communication between a speaker and a receiver.

Semantic assignment is established by the listening agent, which hears a statement of outer speech. To understand the speech statement the agent needs access to the semantic assignment consisting of the wording that is connected to the entire situational context.

The assignment (see **Table 3.1**) is a mapping vice versa, where the wording from speech is connected with the whole situation, meaning with the CEs. When hearing a word, the agent retrieves the situation that is connected with the word in order to understand the context. The table below shows how the mapping is build up. If the outer speech statement "Eat?" is heard the situation is retreived and the agent knows he has to answer due the fact a question has been postulated. Eat is interpreted together with the whole situation. In case the agent is hungry he will formulate a "Yes." in return.

Situational Context	Contextual Entities	Outer Speech
Context Welcome 1	Bella, Agent, House, Near Bella, Move, Joy, Relax	Welcome
Context Eat 2	Bella, Agent, House, Near Bella, Move, Nourish	Eat?
Context Others 3	Bella, Agent, House, Near Bella, Move, Nourish	Others?
Context Known 4	Bella, Agent, House, Near Bella, Stand, Joy, Aggressive	Known?
Context Yes 5	Bella, Agent, House, Near Bella, Stand, Joy, Aggressive	Yes

Table 3.1: Semantic Assignment

It means for each speech statement a concept is stored in the ontoloy in the secondary process so that it can be retrieved for speech comprehension. In summary, the ability of hearing is situational context based, depending on the situation the listening agents links to the wording and moreover it depends on the concept the agent has stored for the uttered speech statement.

Furthermore, semantic, situational context should not be confused with the term common ground. The agents do not act based on common ground, but they interact based on a situational context that has been established from the speech utterance.

For outer speech the process is similiar as for inner speech. As seen in fig.3.5 the precondition is that situational context is identified. Then according to the greedy mechanism the appropriate context entity is retrieved from all five context entities. The importance of a greedy algorithm is that different types of performances can be implemented in a flexible way.

If two agents detect the situational context at the same time only one can utter a speech statement and then the other agent reacts on it. It is not possible that both of them speak at the same time. As for this process, the agent utters the outer speech statement, who is in greater need. Meaning, who becomes an increase in the pleasure level if he/she utters the stament at first. Another point for starting the simple interaction between two agents, is that the current model only supports one acting agent and another motionless agent. The agent, which is able to make actions will start with the utterance.

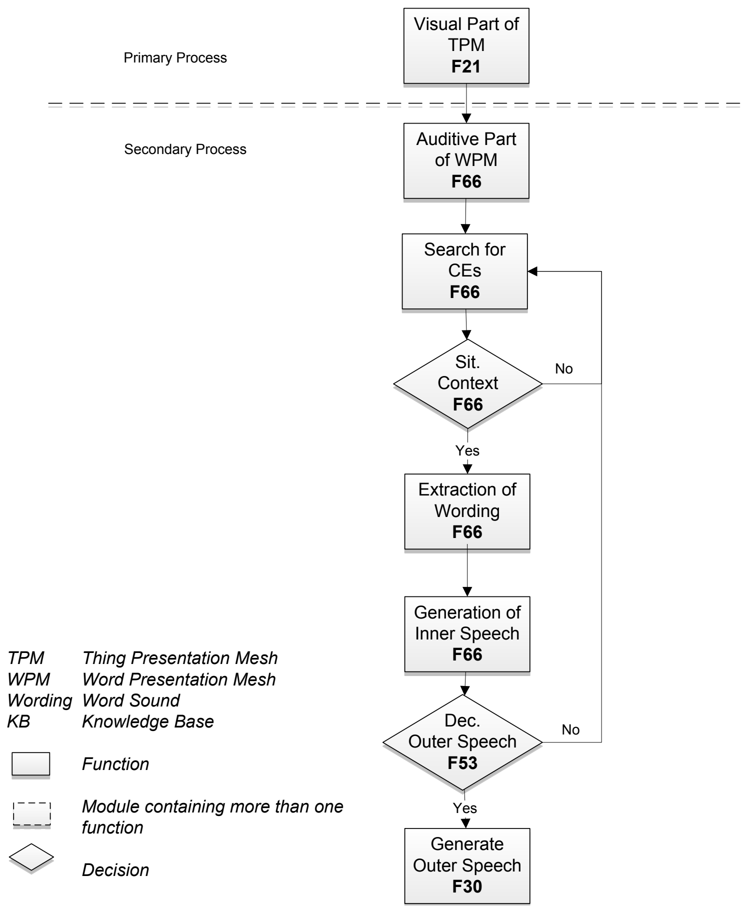

Figure 3.5: Process of Inner and Outer Speech

3.5 Development of Listening to a Statement

The next process described is the listening process, which is discussed in the following. As already shown in the fig.3.7 the listenining depends upon a few factors. Primarily, the speech statement needs to be recieved from the agent, which depends on the fact that an agent is near the other agent. As in vision also in acoustics there is a radar which is responsible for the hearing ability. As the statement enters the acoustic radar, the agent can distinguish among near far and center. Thus, fifteen different types of hearing can be distinguished. As seen in fig.3.6 below two agents can hear them if they say something. The necessary requirement is that their acoustic radars are overlapping.

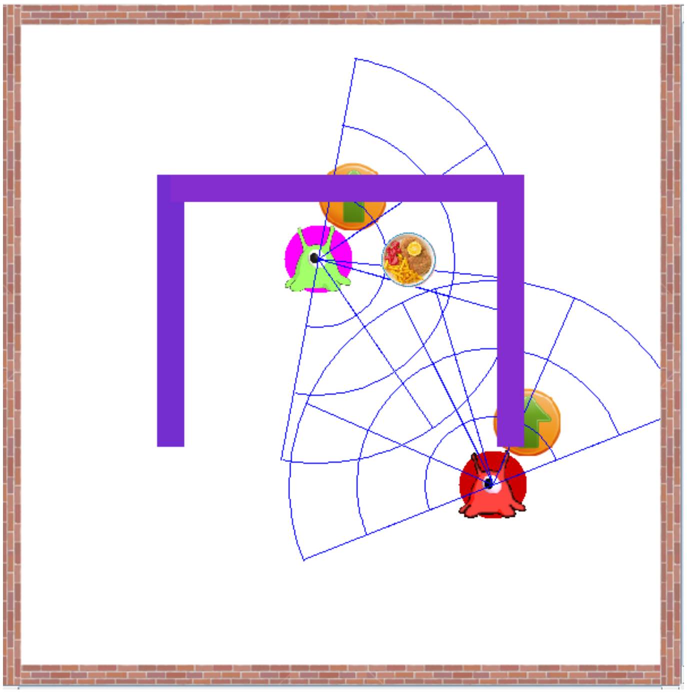

Figure 3.6: Simulator Showing the Acoustic Radar, Agents, Food Source and the House

However, for the process of listening, the speech statement is recognized as belonging to the other agent, due to the data structure: *Acoustic expression*. After it is received the context is set, as the speech statement is part of the situational context now. This means, the outer speech statement becomes a contextual entity. For example, the speech statement Eat will be set as a precondition for the next action. Depending on the configuration *CE: Eat* is stored in memory, the agent can generate yet another output, which is either outer speech or inner speech. It means that the cause of certain situational context can be inner speech or outer speech.

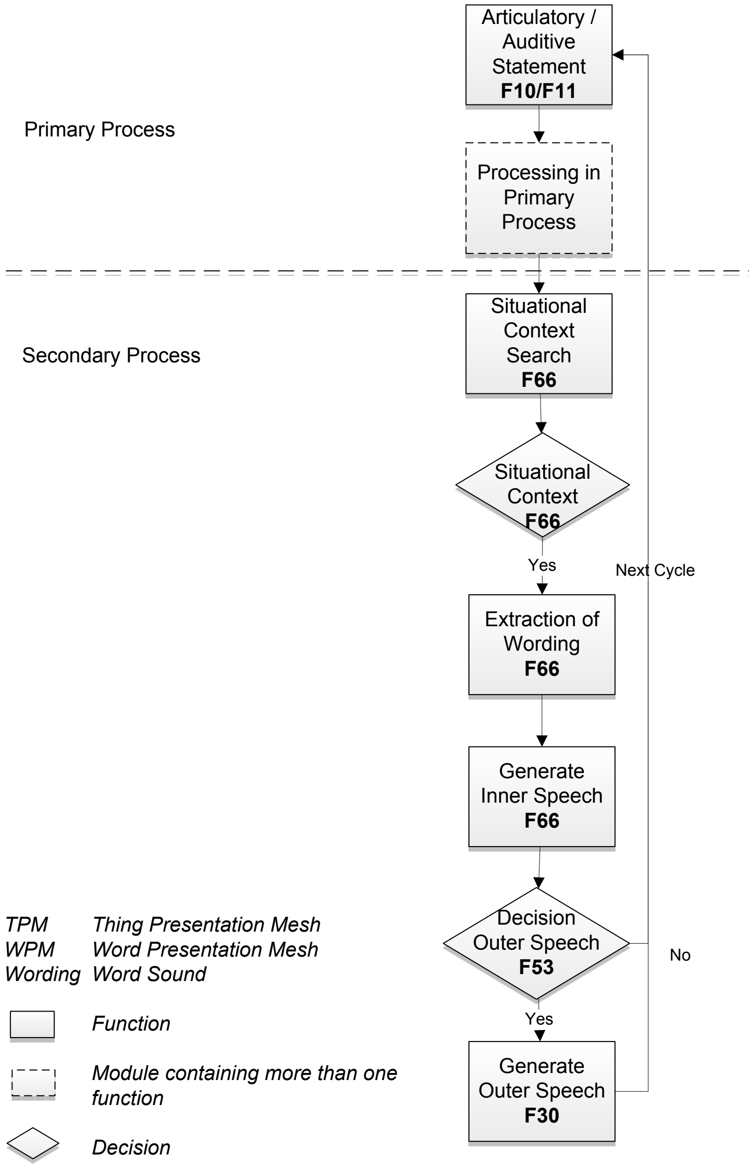

Figure 3.7: Listening to Speech Statement

3.6 Overview of Data structures in the ARS model

The goal of the chapter is to describe the ontological architecture and the new data structures used. For the goal of modeling inner and outer speech the following data structures are important for the model. Secondly the ontological architecture - general and domain specific is explained. The aim of the implementation is domain-specific context recognition that supports navigation and enables semantic context in language.

The overall architecture of the ARS model is based on an object-oriented, top-down modeling approach. As a result of top-down modeling several different layers have been introduced, such as the neural layer for neurological processes and the mental layer for processes of the second topographic model [Lis09, p.24]. A simplified version of the model is presented and discussed in the figure below. Furthermore, the model differeniates Super-Ego, ID and Ego in the mental layer. The horizontal line is the separation between primary and secondary process; The primary process is a process where data is manipulated unconscious, whereas in the secondary process data is processed and manipulated concsious. Based on these processes different data structures are used in the ARS model. Data parameters (DPs) are required for storing information required for the memory of the agent.

Firstly, according to Zeilinger [Zei10] a thing presentation is a primary data structure that is used in the unconscious process of the model. The quota of affects are attached to the thing presentation and to the drives. Apart from this, a thing presentation mesh (TPM) describes a mesh of TPs connected via attribute associations. An association belongs to one of five association types. The attribute association, temporal association, word presentation association and drive mesh association. A drive demand is received from the homeostasis and represents an unbalanced condition. If there is a temporal change of the drive demand the affect is generated as a consequence. Thus, an affect is defined by the temporal change of the drive demand. The drive meshes are associated with TPMs that represent objects. Objects as such, have an impact on the individuals̀ homeostatic state. Further a word presentation is a secondary data structure that is associated with content from the primary data structure the TP.

Thus, in summary the version 38k of the model as stated in the technical review [DBM+13] contains the following data structures:

Data Structure	Data Type
Acts	unconcsious
Quotas of Affect	unconcsious
Association	unconcsious
Association Secondary	concsious
Drive Mesh (DM)	unconcsious
Template Image (TI)	unconcsious
Thing Presentation	unconcsious
Thing Presentation Mesh (TPM)	unconcsious
Word Presentation (WP)	concsious
Word Presentation Mesh (WPM)	concsious
Wording	unconcsious
Concept	pre-consciousness, unconscious
Contextual Entity	concsious

Table 3.2: Table of Data Structures and Data Types

As seen in **Table 3.2** the data structures are described in regards of their data type. Each data structure has one ore more possible data types. Data types belong either to primary and secondary data structure. Primary data structures belong to unconscious processes in a psychoanalytical context. A primary data structures is a thing presentation (TP), thing presentation mesh (TPM), drive, act, template image (TI). The secondary data structures are classes, such as the word presentation (WP) and the word presentation mesh (WPM). The data structure concept is an exceptionary case as it can be found in the pre-conscious and unconcsious. In addition to general, psychoanalytical concepts, attribute information is stored along with each of these basic concepts.

For example, a WP, such as Schnitzel can be distinguished based on their characteristic attribute information taste associated to it.

Second, the individual knowledge base contains individual memory instances, which represent individual memories of an event needed for common memories. Shared or common memories are needed as two different agents can share one event in memory. Reaction rules are needed for triggering action in the case of a certain event under a certain condition. The four vertical lines are a separation based on the neurological part where organs belong to and the categories of the second topographic model [Lis09]. One data paramater unit in the model is considered as a function unit. As shown in 3.8 the different personality parameters (DP) are presented here from 1 to 14. DP1 is connected to F1 (Seeking System), DP2 is connected to F39 (Sensors of Metabolism), DP3 is connected to F11 (Sensors environment) and DP4 to F12 (Sensors Body). In that way, the other DPs are also connected each to one functional module.

The fig.3.8 presents in an abstract way how personality parameters are distributed in the ARS model. They have access to various functional modules in the ARS model. Data parameters in the ARS model are personality parameters that are constants within an application.

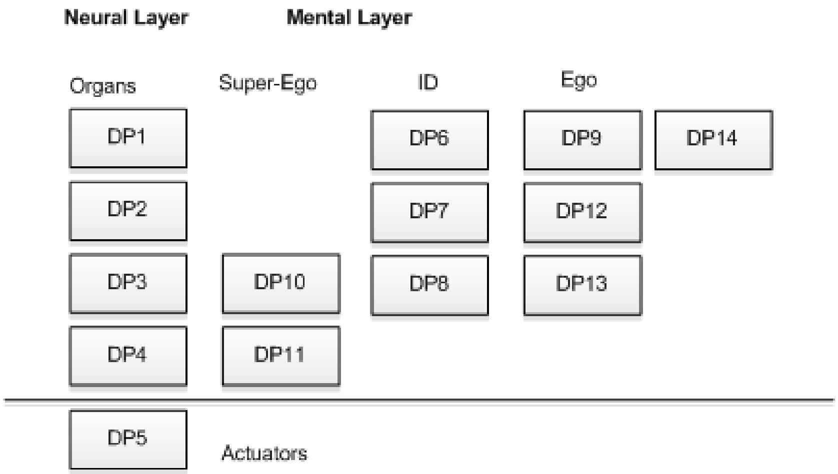

Figure 3.8: Individual Knowledge Bases Mapped Onto the ARS Model

The simulation encompasses virtual agents that interact with their environment depending on their input as perceptions, the ontological structure and functional mechanisms from the proposed language model. Based on the ontological structure it is illustrated what is needed for anchoring semantic context in situation recognition and for language and communciation.

3.7 Development and Integration of new Data Structures

In this chapter the data structures *clsConcept* and *clsDomain*, *clsWording*, *Object Contextual Entity* and *clsAcoustics* are explained and how they are related to existing ones. The fig.3.9 shows the different structures during one processing cycle of the ARS model.[2] These data structures

[2]A processing cycle is considered as one model cycle in the simulation engine, which is a time independent structure.

are needed for the realization of inner and outer speech in the ARS model. The role of the data structures is the storage of speech and speech related information in the different functional modules. Hence, every subchapter describing a data structure explains how the new data structure is related to the generation of situational context and speech generation. Furthermore, different data structures used in the ARS model, ranging from concept to articulatory expression are discussed.

The needed data structures are shown all within one cycle in the model. At first, the articulatory statement enters the perceptional apparatus, which is the first data structure is needed. Secondy, the articulatory statement is transferred to a TPM and a wording, which is actually the structure that accounts for the accoustic content how a word shall be uttered. Third, as long as we are in the primary process the structure needed is a TP or TPM. Then concepts are generated in the pre-consciousness and in the conscious process the context entities with their concepts are needed for identifying the situational context. Within the secondary process words presentations are used as data structure. Last, the words are uttered as an articulatory statement and the cycle starts again after the uttered word comes in as a perception.

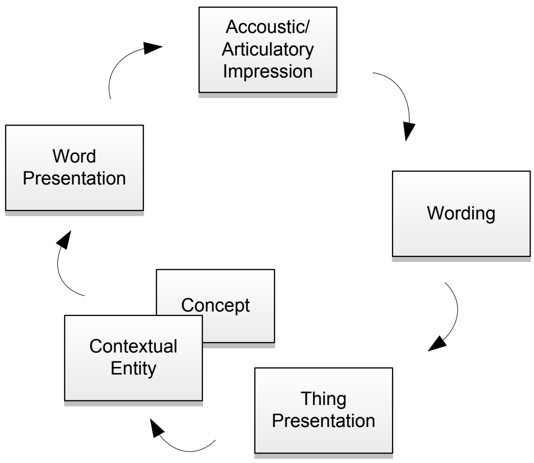

Figure 3.9: New Data Structures Used in the ARS Model Integrated With old Ones

3.7.1 Data Structure Accoustics

The data structure *accoustics* indicates, whether an accoustic input has been received through the perceptional input. This is basically done by the *clsAcoustics* which serves the fact that acoustic perceptions are treated as an acoustic input.

The reason for the split up is that besides a vision expression also an accoustic expression occurs. Thus, when perceptional input enters the model it is differentiated whether there is an accoustic input or a visual input. However, for any accoustic input the data structure is activated and followingly the whole accoustic processing track is activated as well.

In the following the accoustic processing track will be described shortly. At first, a check is done whether an acoustic expression has occured. Secondly, the accoustic expression is transferred into the data structure *wording*. A wording is the knowledge how to pronounce a word. This wording is attached to it's symbolic value in the neurosymbolisation for the body. The phonetics of the

Wiener Schnitzel e.g. meaning how to pronounce it, is attached to the symbol of the schnitzel. This symbol along with the wording is attached to a TPM in the primary process. The TPM is forwarded in the whole cycle until it reaches the transformation to the secondary process. Within this function, the data structure *concept* is retrieved by the wording and TPM. A concept is needed to identify things as they are. Thus, each wording, symbol value and TPM is linked to a concept. Then, the concept is attached to the TPM that is connected to the WPM and from there the appropriate WP is selected to articulate it. When the compound is finally articulated as a WP it becomes a data structure *articulatory expression*.

3.7.2 Class Concept

The data structure concept is meant for all types, where a concept is needed. A *concept* represents a group of entities with the same features [VSZ06]. However in this thesis, the *concept* belongs to the preconcsious and unconcsious process. Added to that, the concept is part of the situational context.

That is in accordance with Tulving, who has characterized semantic memory as being related to noetic consciousness. The term *noetic* is a scientific expression and means that an individual is aware of and thus can cognitevely operate on objects and events [Tul85]. [3]

In this thesis, a concept can belong to one of these five categories, that are derived from the context entities.

- Object
- Emotion
- Action
- Distance
- Drive

The concepts in the framework of this thesis mean the following:

- Category: Object describes objects and persons in the ARSIN world, such as Bodo, Bella, house, Wiener Schnitzel.
- Category: Emotion describes the own emotions of the agent.
- Category: Distance describes, whether an object is far or near.
- Category: Action defines the type of action that is done, such as eat, move forward.
- Category: Drive component.

The content of the concept is retreived from the word presentation meshes, as it can be seen in fig.3.10. The figure shows the representation of a concept in the concept ontology. For the generation of inner speech and outer speech, the concept is specified along with the CE (context entity).

[3] The terms anoetic and noetic have been used by Stout (1896) in a different context.

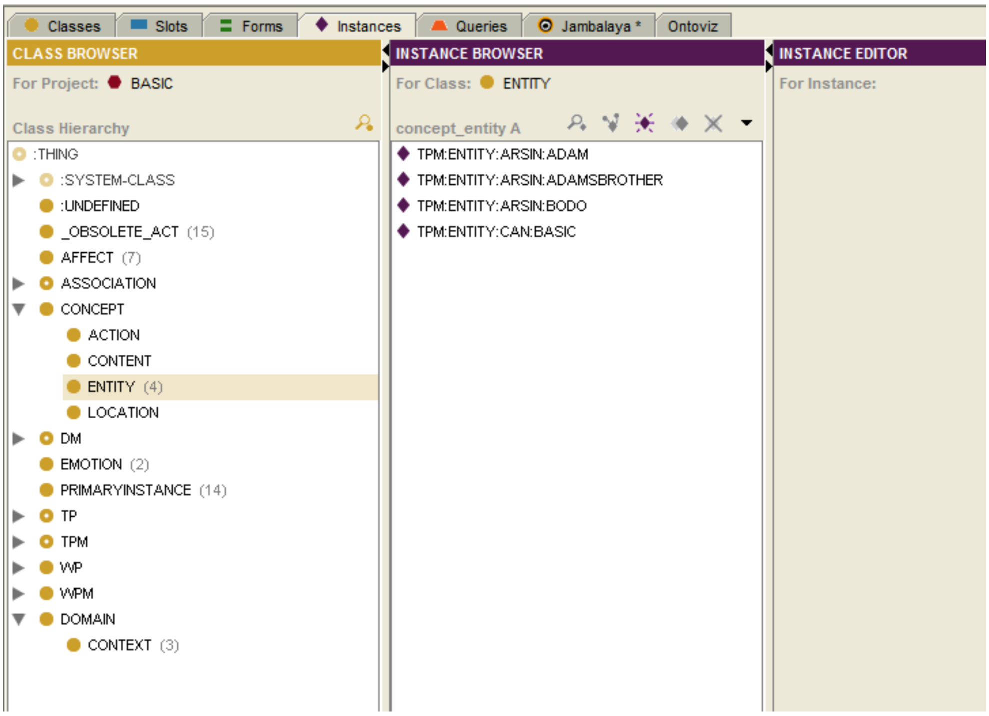

Figure 3.10: Representation of a Concept and it's Instance

One of the main reasons why the data structure *Concept* has been developed was, because the speech production model according to Levelt implies that concepts and contextual knowledge is known before speech is generated. Further, the new data structure is required for the representation of the contextual entity in the primary process. In contrast to the CE the concept belongs to a primary process due to the fact that concepts can not be related to each other and in this work can not have any attributes.

3.7.3 Class Wording

Class wording describes the program class that is used for storing the wording and methods that are associated with the data structure. The data structure wording in the primary process is used for the accoustic content in terms of pronounciation. It is derived from an accoustic expression. The wording contains the activation for an accoustic input plus the information how to pronounce a certain word. The wording in the next step is transferred to the TPM for further processing in the primary process.

3.7.4 Class Contextual Entity

The data structure contextual entities has been made to define and identify a given configuration. The identification of context is based on these five CE's (Person/Object, Location, Emotion, Action and Drive). When they are known as a configuration, the context can be compared to the domain ontology.

In the fig.3.11 the contextual entity is introduced. It is used as an information container in the secondary processes, whereas the concept is more on a primary process level.

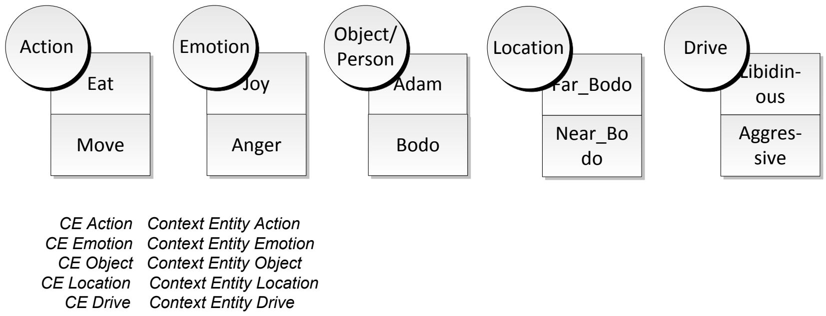

Figure 3.11: TPM Consisiting of Five Contextual Entities and Matching Concepts

Thus, CEs are derived from a TPM and related TPs in order to identify situtational context. A function converts TPM content into contextual entities short before the secondary process. This function will be described in more detail in the chapter *Implementation*.

3.7.5 Data Structure Action Executor

The articulatory expression or message resembles the output of the actuatators in the end. The articulatory expression is also refered to as *speech* or *outer speech* in this work.

For example, an articulatory expression can be sent by speaker in a dialogue. Depending of the perspective of the agent, the articulatory statement is always sent by someone, whereas the acoustic expression is always received by someone. Thus, the articulatory and accoustic expression is needed to distinguish sent speech messages from received messages.

An agent uttering an articulatory message can not send two articulatory expressions at the same time, because only one expression at the time is processed and speech can be seen as an action as well. For receiving a speech message the same restriction is proposed.

Thus, the articulatory expression can be seen as a flag that is attached to the speech statement to know for the agent that he has been sending a speech message. The reason for the mechanism is to certify that a speech message is recognized as one of its own.

3.7.6 Assumptions for Linking the ARS Model to the Speech Production Model

In the this subchapter the underlying assumptions will be explained that are needed for the integration of the ARS model to the speech production model of Levelt.

First Assumption Situational context predominates semantic context. As it can be seen in the process diagram for speech and thought, situational context comes first. The reason is that situational context is concrete and semantic context is rather abstract. Basically it means that before speech is produced in a certain semantic context a situation has to created that enables speech - the situational context namely. It makes sure that the environment is stable. A situation needs to stored in the ontology by all five context entities and the pleasure that is assumed to be gained when speaking out the word.

Second Assumption: Inner speech and outer speech behave according to the reality principle. It means that only if the TP brings pleasure it will be produced. According to psychoanalysis speech is created, if pays off to say something. In case the expected pleasure is high enough, planning comes in and calculates the pleasure for uttering somehting. Meaning, someone only talks, if he/she perceives pleasure out of talking. Therefore, a speech utterance has to generate pleasure according to the pleasure principle. The same is true for the thought process. Thoughts concern the actual perceptions where one is occupied at the moment. For example the actual perceptions when passing objects.

Third Assumption: Semantic context can only occur after inner speech and outer speech have been produced. Semantic context is a process that can only occur in an interchange of at least two agents in a certain situation. It is dedicated to language and overt speech. Semantic context implies that people speak of the same, without explaining themselves. Further, to establish semantic context agents should not move when situational context is established. The situation has to be defined for both agents in order to speak about common terms.

Fourth Assumption: Inner speech and outer speech are one word statements. It means that speach is does not consist of sentences but one word construction. This is based on the fact that inner and outer speech are generated from word presentation meshes and word presentations.

Fifth Assumption: Based on one situational context configuration several follow up actions in speech are possible. One situational context can cause several actions to take place. For example, the situational context Wiener Schnitzel, Bodo and Bella can have multiple turns depending on the psychic content of both. If they are both relaxed they can share the Wiener Schnitzel. If either of them is more hungry the situation can end with one agent eating the food source before the other one arives. Another option can be that Bella asks Bodo if, she can eat the Wiener Schnitzel. Or she might ask whether they can share the food. However, which alternative takes place depends on the psychic content of both agents, even if the situational context configuration has been the same for both of them.

Sixth Assumption The concept of the CEs have to be known by both agents. Otherwise common ground can not be established and both agents are not talking of the same concept. For example, Wiener Schnitzel as a concept is grounded as a concept in the same way for all the agents in the simulation.

3.8 Organization of the Ontology

The ontology contains a general, upper ontology and a domain-specific ontology. These two different ontologies are needed to maintain concepts and store domain-specific contextual entities (CE) for the establishement of the situational context. The separation into two different ontologies is not made within the knowledge base. Hence it is more a structural differentiation that is to be made.

As the organization of the ontologies shows already, the domain-specific ontology depends on an abstract concept ontology, as concepts are fundamental to establish context within a domain. Thus, a data structure domain is needed to manage context units.

In addition the domain-specific ontology is needed for storing CEs.

3.8.1 Organization of the Context Ontology

The organization of the general ontology contains five categories that are needed for recognizing the main situation as such. The question may be asked, why exactly these five CEs have been chosen. It can be explained by the fact that in order to recognize the most basic situations these CEs were appropriate. The approach is however flexible enough to be extended at any time. Meaning, if the situations get more complex in the simulator, the amount of CEs and hereby the context frame can be extended.

The knowledge of concepts is required as a basic for outer speech and inner speech. A *concept* is a group sharing common features among the group as mentioned before. For example, in a situation, where the agent stands in front of the house and observes the food source Wiener Schnitzel. The fact that the agent can recognise the concept of Wiener Schnitzel, of the house and of the other items, as such is due to the concepts.

Crucial for this concept knowledge is that it is general enough for the recognition of situations that is needed for production of a speech message. Without knowing the concepts, speech and communciation is not possible, because the object's concept can not be known. Knowing the concepts is important in speech also in terms of common ground. It is assumed that the agent know their concepts and they are the same.

In the fig.3.12 below one can see the architecture of the general context ontology in the database. The view is called entity relationship model [4] and shows the five CEs (properties) with their possible values.

[4] An Entity Relationship model is used in the field of software engineering and is based on a data model to describe a database in general.

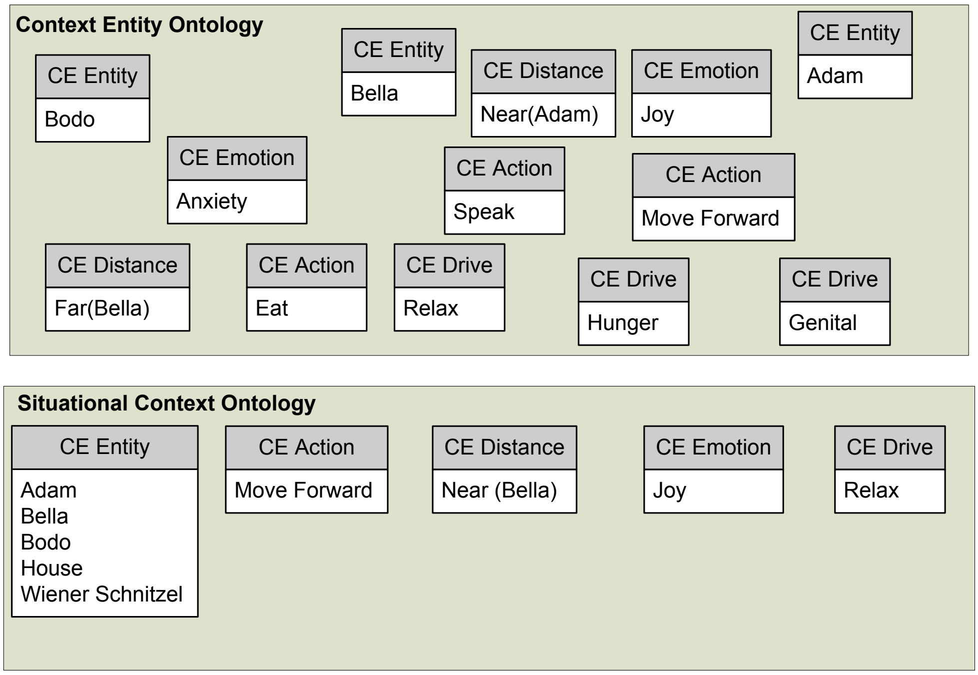

Figure 3.12: Entity-Relationship Model of the General Context Ontology

3.8.2 Organization of the Domain-Specific Ontology

The domain-specific ontology is based on the view of one distinct situation. A situation is already domain-specfic which is why the ontology is called that way.

In the example at hand (see fig.3.13) the domain-specific ontology is based on a situation containing the *CE Entities* Own Pers. Adam, Bella, Bodo, House, Wiener Schnitzel. The additional word "own" indicates that the situation is from Adam's view. The "CE:Action" contains the value "Move forward", because Adam is moving forward. *CE Distance* contains the value Near(Bella). The value near or far requires to set a person or object in the context to know the reference point. *CE Emotion* is set to Joy, which means Adam is joyful. Finally the *CE Drive* complements the information pentagon. It contains the drive of Adam, namely that he is relaxed at the moment.

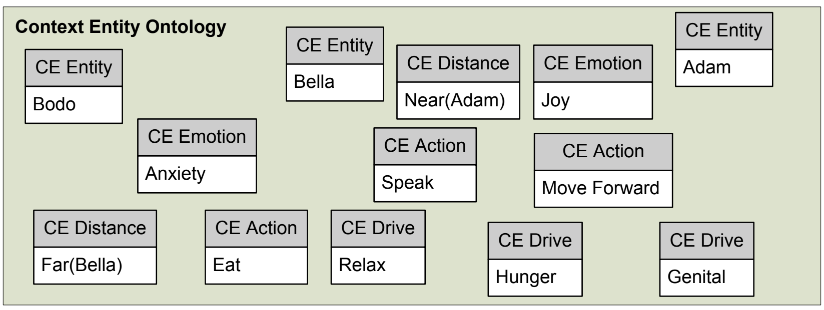

Figure 3.13: Entity-Relationship Model of the Domain-Specific Ontology

In fact, the domain-specific ontology defines situational context. Only after the situation is identified, the overall situational context is defined as well. Based on this only certain speech statements can be selected as appropriate for a situation. Exactly in this domain the agent can only say the word utterance: "Welcome", due to the fact that he retrieves the wording in the next step.

In the fig.3.14 you can see the general and the domain-specfic ontology vice versa in order to see how they are connected. In the general upper ontology the database entries are concepts, whereas in the domain-specific ontology the entries are context entities.

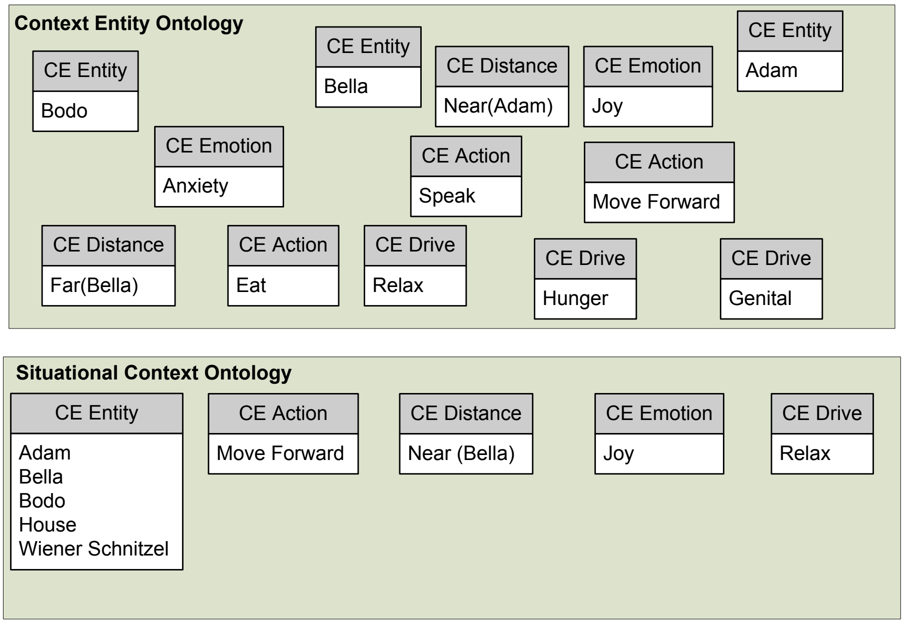

Figure 3.14: Entity-Relationship Model of the General and the Domain-Specific Ontology

3.8.3 Memory Structure in the Ontology

Whereas the general ontology contains all the concepts for all available situations, a domain ontology describes already a situation an. These can be called a set of concept or configuration items. Thus, the data structure domain is needed for storing domain-specific data of all five context entities.

A domain can have two different outcomes.

- Situational Context, and
- Semantic Context.

In order to achieve a) all five CEs mentioned above have to be filled with sensed information from the perception track. After this information is processed further and compared to the situational memories of domains stored in the ontology it is concluded, whether a situation is known or not.

Information, such as a specific context is stored with the name of each instance. The figure 3.15 shows, how the instance *Context Approach for Food* is stored in Protégé. Protégé is used as a tool for storing the memories and concepts of an agent. The instance of a domain encapsulates the CEs: Adam, Cake, Move Forward and Nourish. The appropriate action for these situation is thinking. Meaning the action will not speak in the situation.

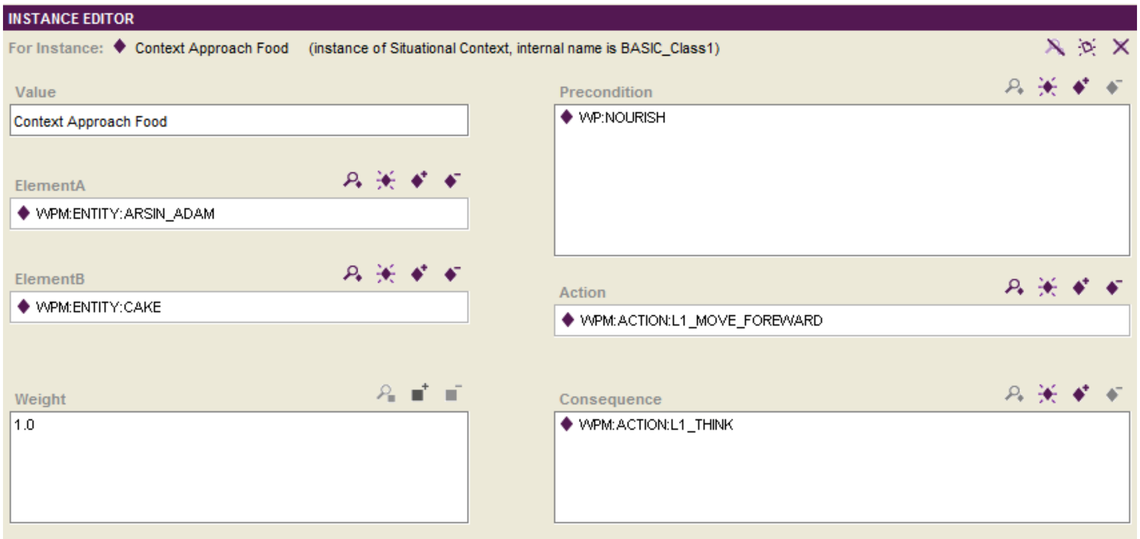

Figure 3.15: Context Approach for Food in the Domain Ontology

By means of the data structure domain, the content for the domain-specfic ontology is defined. A domain consists of concepts from the above mentioned data structure concept. The class *Domain* is needed to describe the overall context as such, whereas with the class concept only items which are part of an context are settled.

The next paragraph describes a situation context for the situational context “Ask for Food”, which is defined in the Protégé ontology as such. The situational context consists of the following properties. Properties are items that are defined in the ontology besides classes at the top level. Each situation has a contex name, then the five CEs follow, last but pleasure level is defined for each situation. It is needed to for each situation as well, for the calculation of the pleasure level.

- Context Name,
- Action,
- Emotion,
- Distance,
- Objects and Persons,
- Drive,
- Action.
- Pleasure Level.

The situation name covers the name of the situation as for this: "Ask for Food". The next five items (Action, Emotion, Distance, Objects and Persons, Drive) are equivalent with the CEs that have been presented in the last subchapter. In addition the *CE:Drive* is needed for setting the action in if-then condition style. The action contains the speech message that is uttered. It is set due to the context of the situation. A weight stands for the amount of pleasure a situation will bring. The weight of 1.0 indicates that the pleasure level is high, if the agent produces the word.

Moreover, the context "Ask for Food" can have different meanings. The situational context can be identified, but there is a chance it can't be identified. Meaning, if context is not recognised then the spread activation starts looking out for all possible associations to objects or people in the situation. Even if there is no situational context recognised, the agent is able to produce a statement concerning the association in the object. For example, the agent will say something about the house in general. He might also notice the cake and just say cake. In the next model cycle the statement is taken into account and if a situational context is recognised as a whole the agent can again answer according the situation. 665 Thus, spread activation statements are gap fillers as long as there is no real situation context. Still the agent is able to produce something in this case.

4 Modeling of Speech Production

"For we have the experience of ourselves, of that consciousness which we are, and it is on the basis of this experience that all linguistic connotations are assessed, and precisely through it that language comes to have any meaning at all for us." Merleau-Ponty

The chapter introduces Levelt's three layered model of speech production [Lev89, p.89], whereby the different layers, as well as macro- and microplaning are addressed further. The model of Levelt assumes first a conceptualization layer, which will be compared to the concept generation process in psychoanalysis. The reason for this chapter is to understand how Levelt's model has been integrated into the psychoanalytical framework. In order to achieve an integration one needs to understand the process of concept generation in Levelt's model and in psychoanalysis. Finally, the neurocognitive perspective is taken into account, as a lot of neuronal processes such as spread activation have been adapted for usage in the model.

As explained in the previous chapter *Concept* the development of situational context is tied narrowly to speech. Situational context is a prerequisite of language and speech - the decision for this step is explained in the following. As speaking requires psychic energy, people, who want to speak need a good reason. Namely the situational context. Without this context, speech utterances would be uttered in a meaningless situation. Furthermore, a situation becomes a meaning, if the speaker can capture the meaning of the speech statement e.g. word and the listener is able to understand the meaning as well. Speaking is always coupled with a sender and a listener, or in other words a speech message that is sent from the speaker to the listener. Finally, message generation will be introduced in this chapter.

4.1 A simplified speech production model

Apart from the model of Levelt several other speech production models have been introduced in chapter *State-of-the-Art*. As it can be seen in the evaluation already, Levelt's model enables the prodcution of outer speech (and inner speech) in a functional model. On the one hand *outer speech* is defined in the Levelt model as an articulated statement. On the other hand, before something is articulated as a word, it is called *inner speech* in his model. Inner speech is a prerequisite of outer speech and both (inner and outer speech) develops from an intention in the beginning of a speech act. Meaning, the intention is the trigger for the speech act. Besides there is an inner and outer loop for monitoring speech utterances during thinking and speaking. This represents the so-called positive feedback in the technical sense.

It is important to underline that the model of Levelt has not been designed originally for speech comprehension. The understanding of speech as he defines it. Levelt claims that speech comprehension, the assignment of semantics to words in the lexicon [1], is realized by a separate speech comprehension unit that becomes as an input the phoentic plan and sends as an output parsed speech to the monitoring submodule in the conceptualizer unit.

Clarification of the concept

"Comprehension of speech" in this thesis means the understanding of words as they are used in the one word sentences. Therefore, the meaning of the word presentation used. The word is grounded with its word presentation and its context. The expression grounding refers to the process of anchoring the semantics or meaning of a word. How can meaning be attached or how does the grounding happen? The grounding of a word is based on the semantic assignment which happens in the ontology used in this work. In this ontology the word is associated to a set of CEs context items. By means of this assignment the grounding of the word takes place and the word is comprehended due to the assignment of the word to the context. Access to the semantic assignment of words is granted by the ontological knowledge base.

In the fig.4.1 on the next page the speech comprehension system along with the modulary speech production system (conceptualizer, formulator and articulator) is outlined. The structure is from Levelt's speech production model. Apart from this model, other structures exist to model language which have been introduced in chapter *State-of-the-Art*.

Conceptualizer: The conceptualizer is responsible for the *message generation* based on factual knowledge and situation knowledge. In addition, there is a loop for monitoring the outer speech. In case there are detected concepts in outer speech that do not fit in, the monitoring will detect them and in the next cycle the message generation is adapted according to the appropriate concept. The output of the conceptualizing is the preverbal message that is sent to the formulator.

Formulator: The formulator is responsible for the grammatical and phonological encoding. Also the message is sent to the lexicon, where lemmas and forms are extracted. The first result after grammatical encoding is the surface structure containing the grammar and the second result after phonological encoding is the phonetic plan. Before a phonetic plan is articulated it is sent to the speech comprehension unit for monitoring. The loop monitors errors of language.

Articulator: The module receives phonetic plan and produces the overt speech that is articulated. The articulation unit is connected with the audition unit as a prestep before speech comprehension is done on the articulated word. The speech prodcution model of Levelt assumes speech comprehension as an extra module that is separate and furthermore linked to the lexicon, where the words meaning is stored. The lexicon in fig.4.1 resembles the ontological database, which has been mentioned above and that is used in the ARS model for storing and retrieving words. In his work Levelt does not give a detailed explanation how speech comprehension works as he concentrates on the speech production as such.

Fig.4.1 in comparison to the picture in chapter *State-of-the-Art* shows different knowledgebases. For once, in the beginning of the speech process the context entities are compare to the factual

[1] The lexicon consists of lemmas and these are stored in an ontology.

and contextual knowledge. Another supporting ontology is between the formulator and the speech comprehension process. This knowledge base is required for accessing the lexicon information and the information about forms.

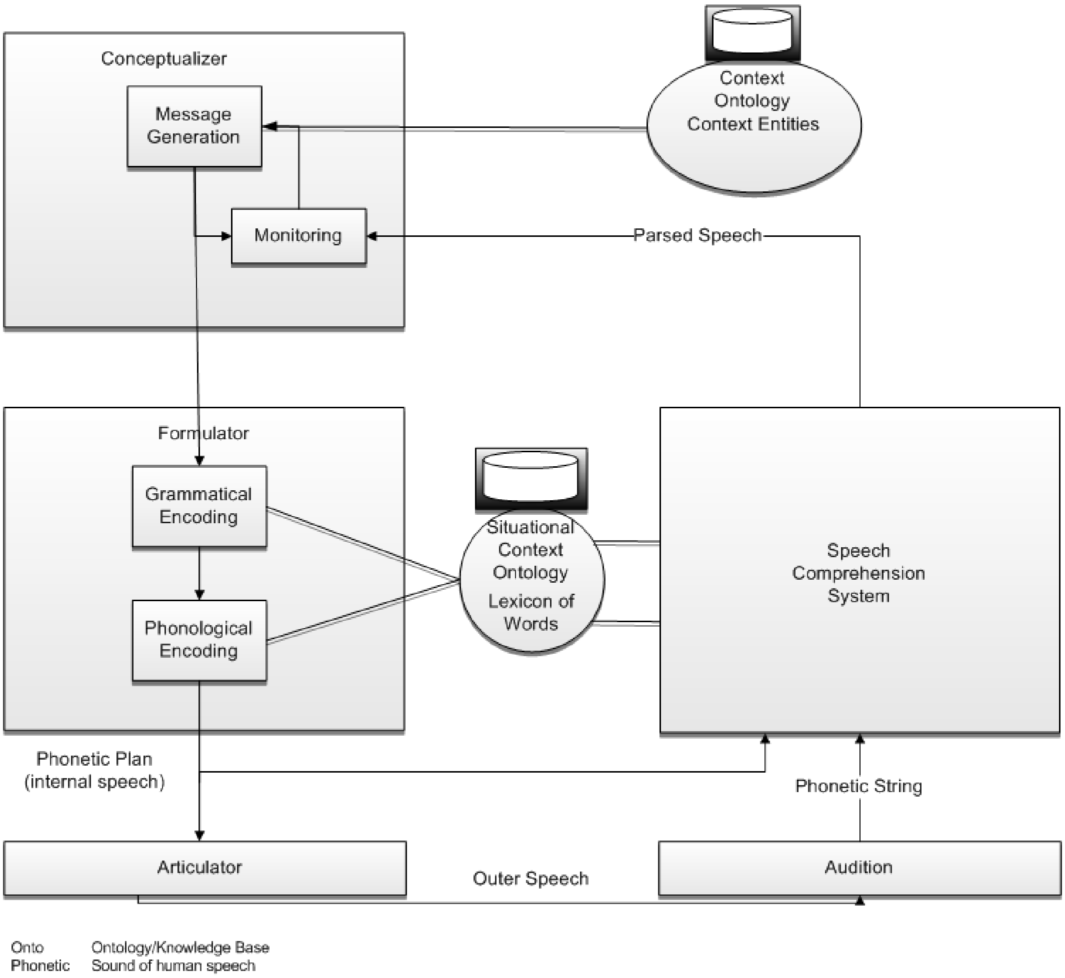

Figure 4.1: Speech Comprehension in Levelt's Model and Knowledge Bases[Lev89]

It should be emphasized that the model of Levelt has been designed to produce inner and outer speech based on intentions. As speech plays an important role, case formation, mood and language dialects are taken into account when treating the structure of the message. As regards this thesis, the mechanisms of Levelt's speech production model has been connected to the functional ARS model. The psycholinguistic theory of Levelt is used for speech production in the ARS model. An adapted version of Levelt's theory is used as the speech comprehension module of Levelt is not taken into account. By adaption it is further meant, that the model of Levelt has to be integrated into the ARS model and thus adaptions need to be done to integrate it.

The following paragraphs illustrates the structure of a message and the preverbal message according to Levelt. The expression means, that it has not yet been formulated as a word. A preverbal message is actually the very first output of the conceptualizer unit. At first the structure of a message is described from the knowledge representation to the preverabal message. The preverbal message is constructed by a cognitive system that communicates internally by means of conceptual codes, such as spatial, propositional and kinesthetic codes. The codes are assumed to be part of the cognitive system. The spatial representation system contains information on transformational procedures [Lev89, p.89], whereas the propositional representation system has information regarding relations between concepts. The kinesthetic representation system is based on senserelated information such as taste, sound or smell. All mentioned forms of representational information belong to the so-called declarative knowledge.

Semantic or contextual entities are structures, where human mind organizes the experiences into defined categories. Concepts, such as persons, objects and events are categories in the real world. In the model proposed the contextual frames for the categorization of situations are the contextual entities: entities, actions, positions, emotions and drives. According to Levelt [Lev89, p.107] different kinds of messages exist. He identifies the following types of information with the categories person, thing, event, action, state, time, place, direction, attribute and manner. The ARS model differentiates the categories person, object, action, relative location [DBM+13] to name only a few. As one can see, there are several overlapping categories within these two models.

Speech statement production (e.g. Eat) in Levelt's model is integrated in the ARS model by using one word to three-word utterances. The reason for this is that the ontology access at hand is too low perfoming. Another reason is, that the whole ontology is very light weighted in term of the amount of concepts in there. This problem can be overcome by manually updating the ontology and entering more words such as prepositions and reflexive pronouns.

Thus, an outer speech statement consists mainly of only one concept or a word presentation. In the case of the statement "Eat" the underlying concept is Eat. In case three-word sentences are produced the whole statement is processed as one phrase and can not be separated. For example: "Is he known?". Then the concept is if someone else is expected, which is as a quite complex concept described in the ontology.

Psychoanalytically, a message is produced as outer speech, if a speech statement is able to create pleasure, according to the pleasure principle [DBM+13, p.96]. That's why, for each concept in the ontology a pleasure value is stored. It means, if a statement is uttered, then the pleasure level is increased by this value. Through this the agent can have more than one possible statements to produce but he will only produce the one which offers a higher increase in pleasure. An example of one situation having two different pleasure levels is given below in **Table 4.1**.

Situational Context	Outer Speech	Concept	Pleasure Level
Context Welcome 1	Welcome.	Greeting and asking someone to come in	0.8
Context Welcome 2	How are you?	Greeting statement	0.4

Table 4.1: Semantic Assignment

The generation of a speech statement depends on the previously established situational context, the inner sppech presentation and the context overall. If the situational context is known or identified, the agent produces the proper statement in a certain situation. The utterance is a one word statment as well, that has the type of an action. The situational context representation is stored as a memory and from the representation the relevant messages can be retrieved.

Fig.4.2 shows all steps of the generation of a speech message in a diagram. In the following the required steps will be explained in detail:

- Macroplaning: determines the content of the speech act and decides what information is expressed,
- Microplaning: shaping of each speech act and ordering the information for the articulation.

The first requirement after the intention for an information to be expressed is the instrumentality of changing the common knowledge. Meaning not every detail of a discourse is conveyed within a speech statement. Due to the fact shared knowledge exists (memory) the listener can infer intentions. Another rule within macroplanning is that information, which can be inferred, because of knowing it, should not be expressed as a speech statement.

Macroplaning is concerned with attention processes as well. The speech statement conveyed by the speaker has to exist in the speakers' focus. This is an attentional process that is solved by the amount of attention that is given. However, in the ARS model the amount of sensory perceptions is limited according to a body data parameter that sets the attentional processes. Body parameters are used in the model for regulating the individual needs of the agent [DBM+13]. It means that specific body parameters resemble the macroplaning in the ARS model. In order to be integrated a body parameter needs to be created for attention at word level.

Apart from attentional processes, macroplaning involves memory or knowledge search, ordering information, and inference making in the knowledge. The ordering of information for an expression is used for planning the expression. It is used mainly if the number of words in a sentence is greater than one then ordering refers to ordering the number of messages. Inference making on knowledge is done by situational context rules. Meaning, during macroplaning inference rules are applied to attentional processes and to the ordering, for example.

In addition microplaning encapsulates four major points that have been postulated by Levelt:

- Assigning Accesibility Status to Referrents,
- Topicalization,
- Propositionalization, and
- Language Specific Requirements.

The first point, is concerned with the fact that a speaker has to introduce persons and events. A speaker refers to events and persons in such a way that the listener recognises the concepts during discourse. If the listener does not know the used statement or concept, the responsibility of the speaker is to recognise it and use a known concept.

The second point addresses topicalization, meaning that the speaker has to select the information he will utter in ordert to capture the attention of the speaker. It means actually that the speaker has to make sure what he tells the listener is novel to him so that he pays attention.

The third point means that images need to be assigned into a propositional form which is necessary for the perspective taking process. The final point is language specific requirements which includes routines and procedures for the tense system. These procedures are automatically conducted and do not acquire a lot of attention. At the end of microplanning the preverbal message is constructed that can be seen in the fig.4.2 at the begin.

Microplanning is not fully integrated in the ARS model, as the mentioned points are quite to specific for the rather simplified model. However, for sufficiency reasons and for understanding the process of intention to speaking it needs to be mentioned here.

In summary to the last paragraphs, the fig.4.2 shows the functions of macroplaning and microplanning. The speech statement is produced after the last module in this figure, meaning the whole process has a speech statement as an output.

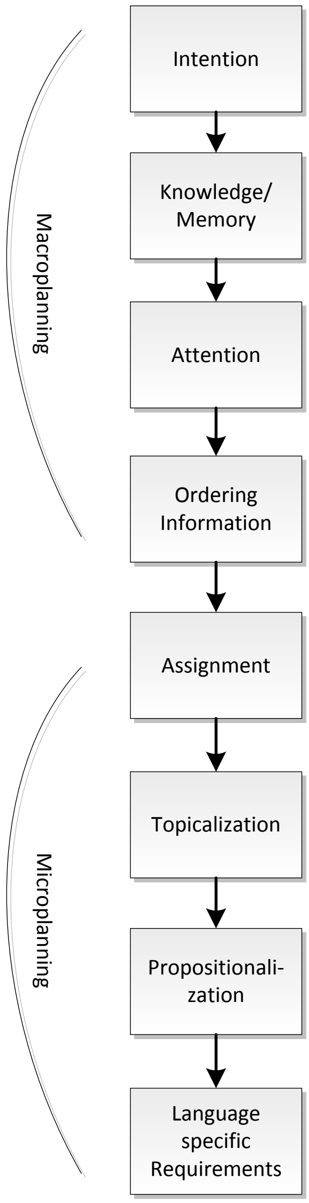

Figure 4.2: Microplaning and Macroplanning

4.2 Concept Formation in Psychoanalysis

In order to integrate Levelt's model into the ARS model one has to know how concepts function in psychoanalysis. That is why, in the following chapter it will be explained what the role of concepts is in psychoanalysis. What is a concept and how can it be integrated with the word and thing presentation from the ARS model. Furthermore, the role if the pleasure principle is illustrated at the end which is fundamental for the functioning of the emotional apparatus.

4.2.1 Psychoanalytical View

The role of concepts and conceptualization in psychoanalysis is laid out here in order to understand the usage of concept from a psychoanalytical point of view.

In general, the role of language and speech in psychoanalysis has been not directly addressed. Indirectly however, some works have dealt with language from a psychoanalytical point of view, such as the thesis of Doblhammer that discusses besides the development of language in childrens' language, the Freudian perspective on language as well [K.98]. Although Freud splits up the representation into a word and a thing presentation, we do not find direct hints for the development of normal functioning speech and language in classical psychoanalysis. Apart from this, the first written work of Freud concerning speech and language was about describing the different types of aphasia [Fre91]. Furthermore, in his letters to C.G. Jung, Freud has tackled language development and lingustics as well, though there have not been made detailled obsverations on this topic [Fre15a].

At first the inner speech process can be described as bringing together different associations. Inner speech processes, such as thinking depend on thing presentations. For example, two agents produce in the primary process both exactly the same thing presentation. However, outer speech is guided by the drive component and the situation. Thus, if one of the agents has different drives due to a situation then the speech statement is of course different.

Thinking, which has in German language the meaning of "Denken", is described in psychoanalytical terms as an ego function, resulting from drive deferral and the processing of mental contents based on the secondary process. To reason means "Nachdenken" in German and represents the following in psychoanalysis. It is a conscious process and describes the relation between word presentations in order to conduct logical operations based on psychic content. Reasoning includes a measuring system as the the quotas of affects already.

The following fig.4.3 shows, how the articulated word presentation manipulates a thing presentation mesh. The different functions belong to different modules, such as the TPM, which is active in the beginning when a word is heard in the acoustic path starting from F10. In F66 the Wording is generated and is processed together with the word presentation. The articulation module takes place in module F30, where the word presentation is articulated on the basis of a WPM. Thus, the wording is attached in the beginning of the secondary process to the word presentation.

By manipulating it is meant that outer speech influences inner speech. This is the case in the Sapir Whorf thesis as well. The Sapir-Whorf thesis is named after the linguists Edward Sapir and Benjamin Lee Whorf and represents two very different perspectives. One is the lingustic determinism saying that thinking is determined by language whilst the other party of the theory claims that different language leads to a different thinking as well [Cha95]. The Sapir Whorf thesis assumes, that language is guided by thinking and thinking in turn manipulates language. It is always a bidirectional relationship between these two concepts.

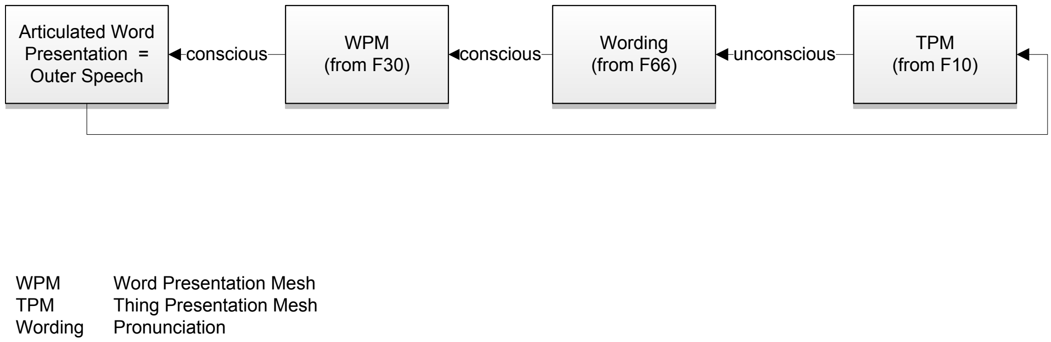

Figure 4.3: Word Presentation System Influencing Thing Presentations

In the thesis at hand the articulated word presentation resembles to outer speech and thing presentations attached to the word presentation to the inner speech in Levelt's model.

Regarding conceptualization, Freud assumes that even an infant is at the beginning of inner speech and concept formation, if it can tolerate frustration. He refers to this epoch as infantile hallucinatory state that is in place until the mother comes. Moreover, he suggests that from this frustration and hallucinatory state the infant develops thinking and conceptualization for the purpose of adaption or modification [HTA82].

Conceptualization in psychoanalytical terms in general means that, if pointing to a thing or an object, not only the signifier is activated, but in addition the concept needs to exist and is activated before the thing- and word presentation is activated. Thus, when speaking to each other one has different concepts in mind depending on what has been received by perception beforehand.

Psychoanalytical Hypothesis's

In the following subchapter all hypothesis from psychoanalysis that are needed for speech are described.

Hypothesis 1: Link between thing- and word presentation. The visual image of the thing presentation is linked to the sound image of the word presentation. The requirement is important for the attachment of visual objects to sounds and furthermore that each sound image is linked to an object in the secondary process. As it can be seen in the fig.4.4 below. The word presentation consists of a acoustic pattern, motion picture, reading image and the writing image. The motion picture represents how something is written in motion.

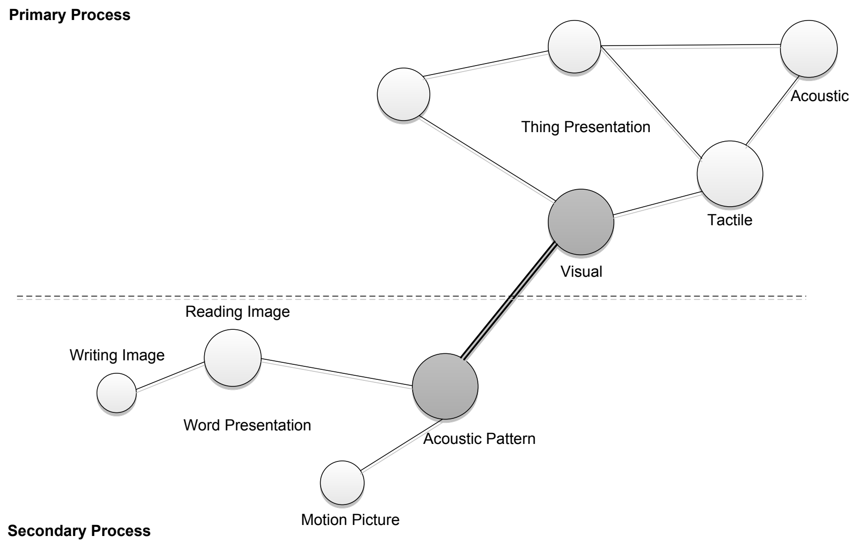

Figure 4.4: Linkage between Thing Presentations and Word Presentations

Hypothesis 2: Inner Speech is equivalent to inner monologue. Inner speech is a preconcsious or conscious usage of word- and thing presentations that are formed in the form of sentences. In this work one word sentences are used like in child language for simplicity reasons. Inner speech takes place in the preconcsious planning and in the form of mumbling words or sentences.

Hypothesis 3: Outer speech is motorically articulated und articulated speech as an action. It is conscious, but can be unconcsious as well in the case of speech errors. Important is that inner speech predominates outer speech.

Hypothesis 4: The Drive Compound is the trigger for Outer Speech. It means that the drive is the motivation for outer speech. Without the drive itself there is not motivation to speak. In addition, the drive compound sets how to say something, which is decdided in the planning modules later.

Hypothesis 5: Pleasure value determines what is actually said. How something is actually said is the socalled act of "Voraussagen". It depends on the pleasure level that is calculated in module F53 for wordings in advance before they are produced. The pleasure level is quite high for a word that pays off to be articulated, whereas if a word that has low pleasure, it is less likely to be selected in a situation. The pleasure level has been added to each concept in the knowledge base.

Furthermore important is, that when uttering a statement, the agent receives a certain amount of pleasure for uttering the statement. For speech content it is decided depending on the pleasure principle in the primary process whether speech content can be passed on. If this is the case the content becomes a WP and can become preconcsious. In the next step the reality principle decides, whether a preconcsious content can be uttered in the end.

Meaning, only when a preconscious content brings pleasure and avoids unpleasure in reality it will be articulated at all. Additionally, producing a speech statement also decreases the drive tension and thus increases positive emotions, such as joy. The reality principle is the reason, why one statment is prefered over the other statement due the fact that one statement can eventually cause more pleasure than the other and the reason why it is uttered at all.

In comparison to the Levelt model, where speech is generated based on intentions, in psychoanalysis speech is based on a 6steps measuring system. The Levelt model of speech production starts with the plan to produce something, which is a much more specialized aim than the need to say something. Thus, to integrate the Levelt model, it needs prior to the intention to say something the need or drive to say something. The gap between those two models is closed easily in the way that need triggers intention. In the proposed model of this work, the need for food and the resulting hunger triggers the agent to say "Eat!" for example. Otherwise the agent would only think about objects that bring the most pleasure to him according to the reality principle.

4.2.2 Neurocognitive Perspective of Modelling Speech and Language

The chapter takes into account existing research in the ARS project in that perspective. Furthermore, it discusses what kind of preassumptions are made from the neurocognitive side that a word presentation is uttered in the end. *Cognitive Neurosciences* deals with neurosciences and cognitive psychology with the goal to investigate the brain and brain or cognition processes.

One part of cognitive neurosciences is modelling, especially the building of cognitive models. As an example, the work of Velik [Vel08] will be mentioned. She described a classical neural network model, using consisting of input nodes, hidden nodes and ouptut nodes. Neural network related processes have been used in the former ARS project. They also have been integrated by means of the spreading activation meachnism that has been used for the activation of associated memories in the work of Wendt [AWMB13].

The following fig.4.5 shows the initially spread activation of Wendt. Template Images (TI) are compared with the perceived image (PI). Associations are created, if the quota of match exceeds a defined threshold. If all components from a TI can be found in the PI, the weight is 1.0 and a direct association is established. Other associations are indirect associations. Spreading of activation in the version by Wendt starts from the perceived image. How psychic energy is spread is decided by the quantity psychic potential that can attract psychic energy [AWMB13].

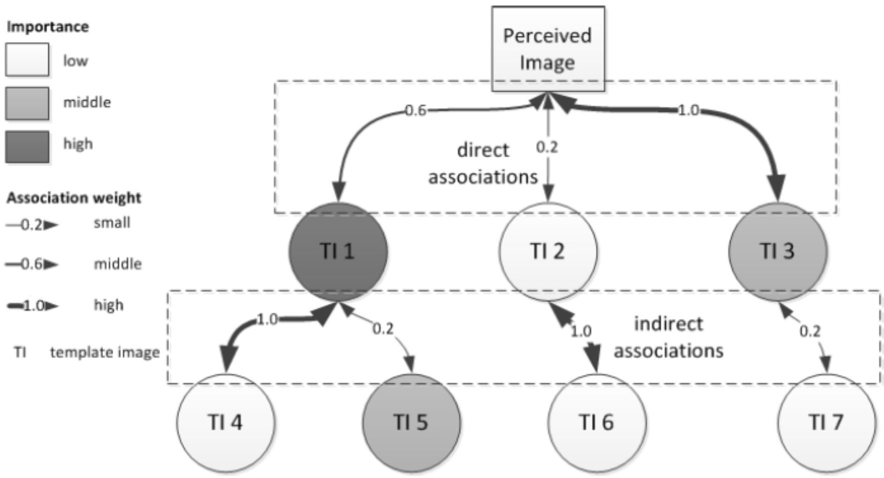

Figure 4.5: Wendt's Version of Reverse Spread Activation

Activation in the case of spread activation goes from the root context node to the bottom. The reason for spread activation in the original version is to establish direct and indirect associations.

Spreading activation is originally a process defined by Dell [Del86] and has been adapted in the ARS project. He defined spreading activation as a mechanisms for accessing memories and spreading the activation from image to image in order to find a certain memor instance.

Speech processes that can be implemented by a neural network are learning of verb flection, for example. The generation of regular and irregular past participles, such as "go-went-gone" is a process that can be done by a neural network, for example [Hin09]. Moreover, the learning of new vocabulary can be implemented based on neural networks due to the fact they are used for learning behaviour. Typically, a neural network has to undergo a number of cycles to learn a new syllable or word. The amount of cycles depends on the structure of the network (e.g. hidden layers) or the damage that as been done to the structure of the network. Typically, language learning in neural networks is represented by an U-shaped curve [KM91] as it can be seen in fig.4.6. In the beginning the learning rate is faster until it stays stable. The stage, where the learning rate stays the same is the bottom of the U-shaped curve. Finally, the learning rate becomes slower, if an over-learning has happened. Thus, the point is, to set the number of learning cycles so that overlearning does not occur.

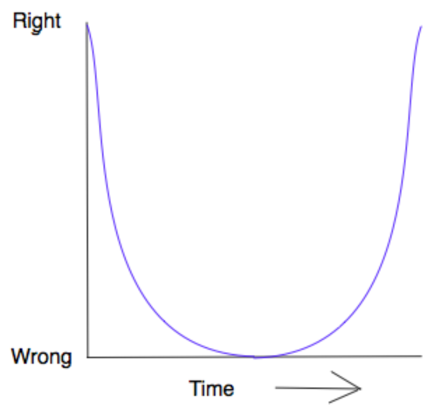

Figure 4.6: U-shaped Curve in Language Learning

In case of simulating deviations in language learning neural networks can be used as well. For example, when deleting some of the nodes the learning is influenced with a great impact. Depending on which nodes are deleted the learning rate can be manipulated a lot. In Levelt's speech prodcution model, Roelofs assumes lemma retrieval in the mental lexicon by spreading activation. Exactly it is done by enforcing the activation level of the node in the network. The activation then spreads from the node towards the lemma level. Then the highest activated lemma node is selected [Roe97].

5 Design Methodology and Implementation

"Only the one who does not question is safe from making a mistake." Albert Einstein

Implemention of speech in agents depends very much on the model that has been used. On the other hand the target group using the simulation defines the granularity of the speech system used. Is the target group interested in agent to agent communication then it might not be needed to share every communication detail with the observer. On the other hand if we observe agents behaviour in a black box type way and want to gain more insight on their individual behaviour it is needed to have more insight to what they think or speak and why they act how they act.

As for the ARS project that has been designed in a top-down way from the general to the more detailled now, the community is interested very much in the detailled condition of the agent's psyche. It is not just about agent's interacting in a psychoanalytical way, but the way how and why they interact is of importance. That's why inner speech and and outer speech enables the agent to comminicate his/her needs.

In this chapter it will be first the design methodology will be explained based on the tools that have been selected. Then the implementation of the speech production model is laid out in a modularized way based on the main modules for speech and thought selection as well as inner speech and outer speech production. Last but not least, the interfaces to these module are highlighted and the implementaion of alternative actions is described.

5.1 Selection of Tools

In the following section the tools Mason, Protégé and Eclipse are discussed, as they are used in this thesis for the development and implementation. For the simulation of the agents MASON is selected, which is a tool for the simulation of multiagent environments and will described firsthand. MASON is used as a libraray for the simulation, whereas Protégé is a database storage tool and Eclipse is a programming environment that can contain the MASON libraray.

5.1.1 Tool for Simulation - MASON

Mason is used in this project for simulating a multi-agent environment, which is why it is described here further. In principle Mason consists of a Java core library called Multiagent Simulation of Neighbourhodds library (MASON). It is a Java [6] library that is popularily used for simulating agents and its neighbourhoods as the name implies. According to Luke et al. MASON is a simulation tool for discrete events. The tool is very practical for agent- and swarm simulations as these kinds of processes are very capacity demanding for a computer[7]. Due to this economic benefit, MASON is used in different fields, such as biology, sociological and economic fields. Originally, MASON was designed as model library where one can easily add features [SLS04]. The advantages of MASON which are used in the ARS project are:

- Visualization in 2D and 3D dimensions,
- Efficient support for as many agents as possible (based on memory),
- Easy connection to larger libraries, such as.

In fig.5.1 the MASON simulator is shown including robots that collect dots in a 2D simulation just as the ARS model. The movie can be downloaeded on the MASON homepage [8]

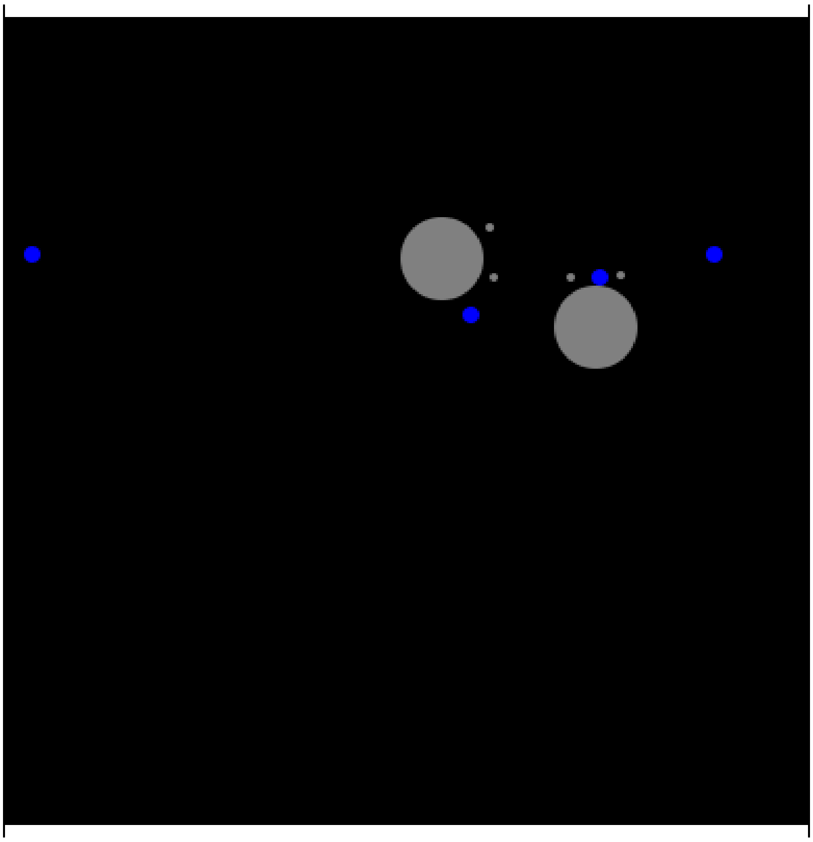

Figure 5.1: MASON Simulator Showing two Robots Collecting Dots

[6] www.java.com
[7] Capacity demand is based on the working memory of the computer.
[8] http://cs.gmu.edu/ eclab/projects/mason/extensions/physics2d/

5.1.2 Tool for Ontology Engineering - Protégé

For the realization of semantic and situational context, a reasonable tool has to be selected. The requirements for the tool are the possibility of storing knowledge and its meaning, accesing knowledge and comparing actual perception information to the data stored in the tool environment. Based on previous works of Zeillinger, [Zei10] and Deutsch [Deu11], they used an environment called Protégé, that has been used for realizing the memory layers of the model. For the work at hand, Protégé is an easy to use ontology editor for implementing an RDF/OWL ontology.

Protégé is an ontology engineering tool that has become very popular in the field of the biomedical Semantic Web. According to Knublauch and his collegues the reason therefore is, that Protégé has been extended for OWL. Originally, the tool has been used for very specific biomedical tasks and developed as a generic tool for all sorts of knowledge based problems. The condition widget permits support of description logics including necessary and sufficient conditions. Another aspect is that description logics follows the open world assumption that concludes: what is not said means a lack of knowledge, we can only talk about things we know and what is not said is false.

An advantage of Protégé is that is supports the generation of Java code from the application. Moroever, until now several packages have been developed in order to visualize the graph layout of classes. By supporting OWL it enables not only modelling of structure but modelling of semantics, which is needed exactly for the task at hand, as we need to model semantic and situational knowledge. Thus, Protégé is used as tool in order to depict situational knowledge in a scene and semantic knowledge during a conversation [HOM04]. It is available in two different versions - frames and slot version, whereby in the ARS project the frames version is used. The current version used is the Protégé-Frames platform in version 3.4.8 Build.

In fig.5.2 an example of the ontology used in ARS is shown, presenting the class THING at the top and the subclasses AFFECT, ASSOCIATION, DM, DOMAIN, EMOTION, PRIMARY INSTANCE, TP, TPM, WP and WPM. An OWL ontology in Protégé consists of Individuals, Classes and Properties. In fig.5.2 below the class window is shown along with the super class :THING, which is abstract. Slots are equal to properties in Protégé. The forms generator is used for generating the forms in Protégé. Instances in Protégé are individuals that are part of a class.

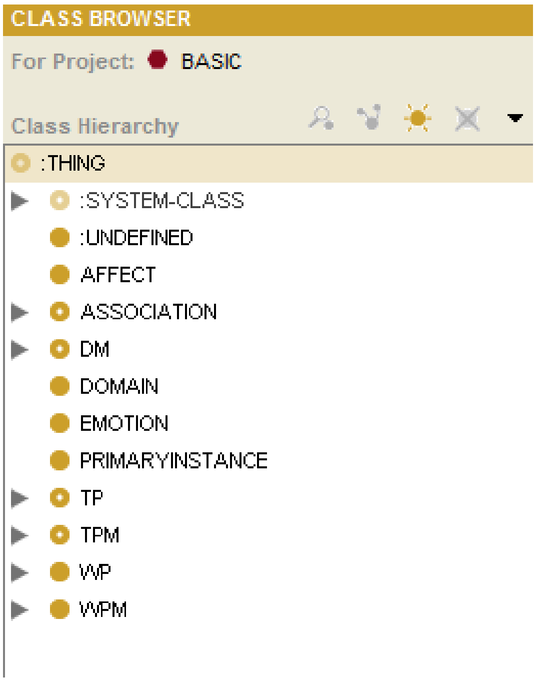

Figure 5.2: Protégé Example

5.1.3 Tool for Programming - Eclipse

The framework eclipse Indigo version has been used for programming the work at hand. The advantage of using Eclipse is the integration of eGIT[9], which has been used as a version management tool previously. In addition, the tool offers multidisciplinary functions for programming, such as package support and integration of other libraries for software sharing.

The eclipse framework supports java developers and offers the possibility of integrating other interfaces, such as Protégé, databases and linked *.xml Files*. An XML file is a file written in eXtended Markup Language. Due to the multidisciplinary functions of eclipse it can be time consuming at the beginning. On the other hand, an advantage of using eclipse is the fact that one can select from a range of packages in the libraray. There might be other programming environments like MS Visual Studio[10] that are practical for the development of Java projects.

5.2 Realization of Situational Context

The categorization of situational context depends very much on two ontologies that are responsible for storing conceptual memory and domain or contextual memory. Protégé is the tool that is used for storing memory traces of the agents. In the work at hand two types of ontological knowledge bases are needed:

- A context ontology or knowledge base for recognizing the context entities,
- A domain-specific ontology or knowledge base for recognizing the situations.

[9]Link provided at: www.eclipse.org/egit/
[10]Link provided at: www.visualstudio.com/

5.2.1 Context Ontology and Context Entities

The context ontology entails all items that are needed for the categorization of a certain situation. A context is defined as the connection between several subunits. The term concept entity is defined as one subunit that is part of the whole context. To recognize a situation all context entities have to be known, which is possible by an access to a knowledge base called protege. Then all incoming perceptions are compared to the situational contexts in the knowledge base. One situational context is exactly the memory of a situation and is stored in the knowledge base as a memory trace.

As seen in the categorization map without the known context entities for Object, Emotion, Drive, Action, Distance the agents is not able to make the categorization and identification.

However, four main contextual entities as listed below are needed before situational context can be developed in the next step. An additional contextual entity is the *CE Drive*, whose role has been described in Chapter *Concept and Modell*. In addition, the CE is described as a part of the situational context.

The reason for these five CEs depends on the categories selected once and second depends on the perceptional inputs and how much one wants to differentiate the input. The number of the contextual items is a measure for the granularity of the situational context. The more categories exist the more finer a context gets. The question when designing such a context frame is how detailed it should be. The number of CEs depends on the amount of perceptional information that is available in the model environment, but it also depends on how the situational frame is set and how much information the agent needs to recognize a situation. In the example below the number of categories is presented.

Basic Categories Selected for Capturing Situational Context

- Context Entity: Object,
- Context Entity: Action,
- Context Entity: Emotion,
- Context Entity: Distance,
- Context Entity: Drive.

Apart from this, each CE has some *property values*, whereby the context entity can be seen as *category*. A value of the CE Object might be Adam as shown in the example below. The *situational context Eat* below is described by the following contextual entities including their values.

Example for the Situational Context Eat

- CE: Adam, Bella, Wiener Schnitzel,
- CE: Move,
- CE: Joy,
- CE: Near to Bella,
- CE: Hunger.

5.2.2 Situational Context Ontology

The situational context ontology is the knowledge base (KB) that contains memory traces. The reason for the KB is that situational context is recognized after a comparison operation onto the KB. The importance is that beforehand the four main contextual entities are required to compare them to the memory traces in the KB. The situational context memory contains the *precondition* and the *consequence*. The precondition is the number of acquired CEs, whereas the consequence is the wording for the speech statement. The wording is transferred to a speech statement that is articulated then. In the next paragraph the structure of the situational context ontology is described more clearly.

The structure of the situational context ontology is simple. The context ontology is derived from the :THING as a whole and contains all the concepts of entities, that are needed in the agent's world. The situational context ontology includes the different situational context scenarios and is also derived from the :THING. Thus, producing a inner or outer speech statement depends on the existance and access of these two ontologies.

Fig.5.3 shows how the ontologies are integrated in the speech production model of Levelt and how it is linked to the functional modules of the ARS model by highlighting the Fx modules. The access to the ontology is given when the lexicon is needed, meaning eventually when the wording is retrieved. Apart from this, the lexicon is needed when a word is heard and the speech statement needs to be looked up in the lexicon. The second ontology can be seen on the top of the image, where the concepts are described. This ontology is the socalled context ontology.

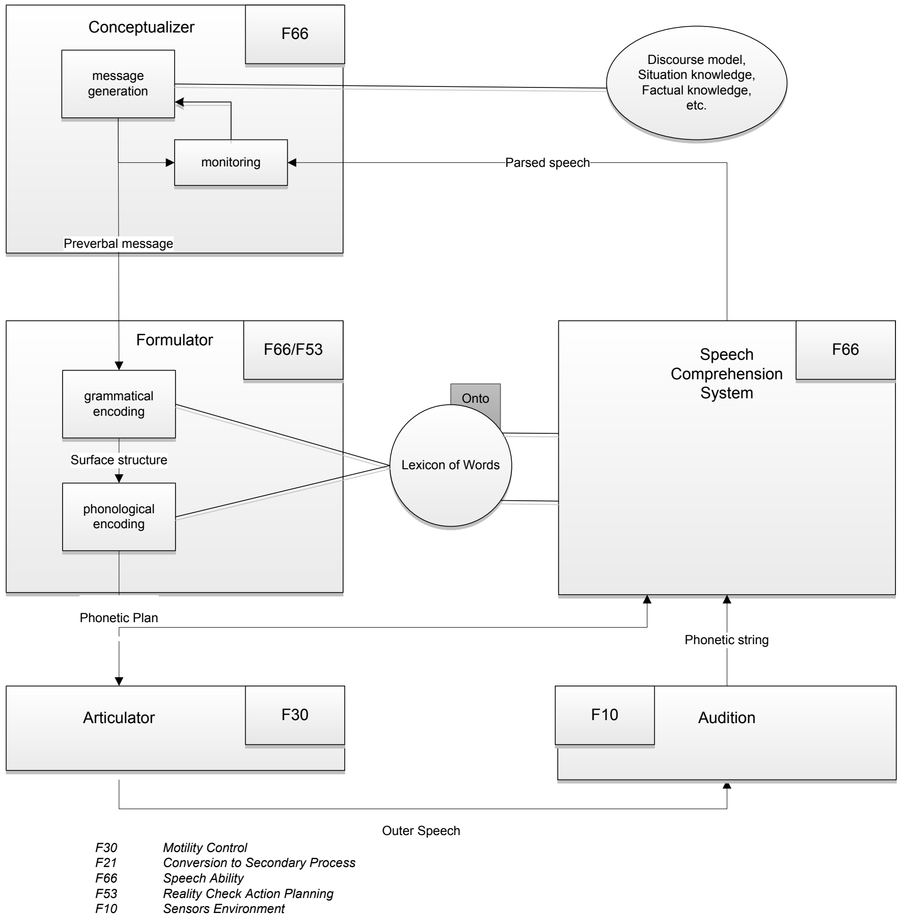

Figure 5.3: Domain-Specific Ontology and Upper Ontology in the Speech Production Model of Levelt

Fig.5.4 in addition shows the different modules in the ARS model and how they are involved in producing speech, which are the purple coloured modules. As it can be seen in the beginning the functional module F21 is responsible for concepts and in the next module F66 the wordings are extracted from the concept. In functiomal module F26 the decision is made on which goal is executed based on goals that have been formed before. F66 is also responsible for speech comprehension as the semantic mapping is done there. Last but not least, the speech is produced by F30 and can be perceived by the modules in the perception F10 and F11.

To see how the speech modules fit in as a whole, the functional model of the ARS project is presented in the full version m in fig.5.4. The modules highlighted in purple colour are involved in speech production.

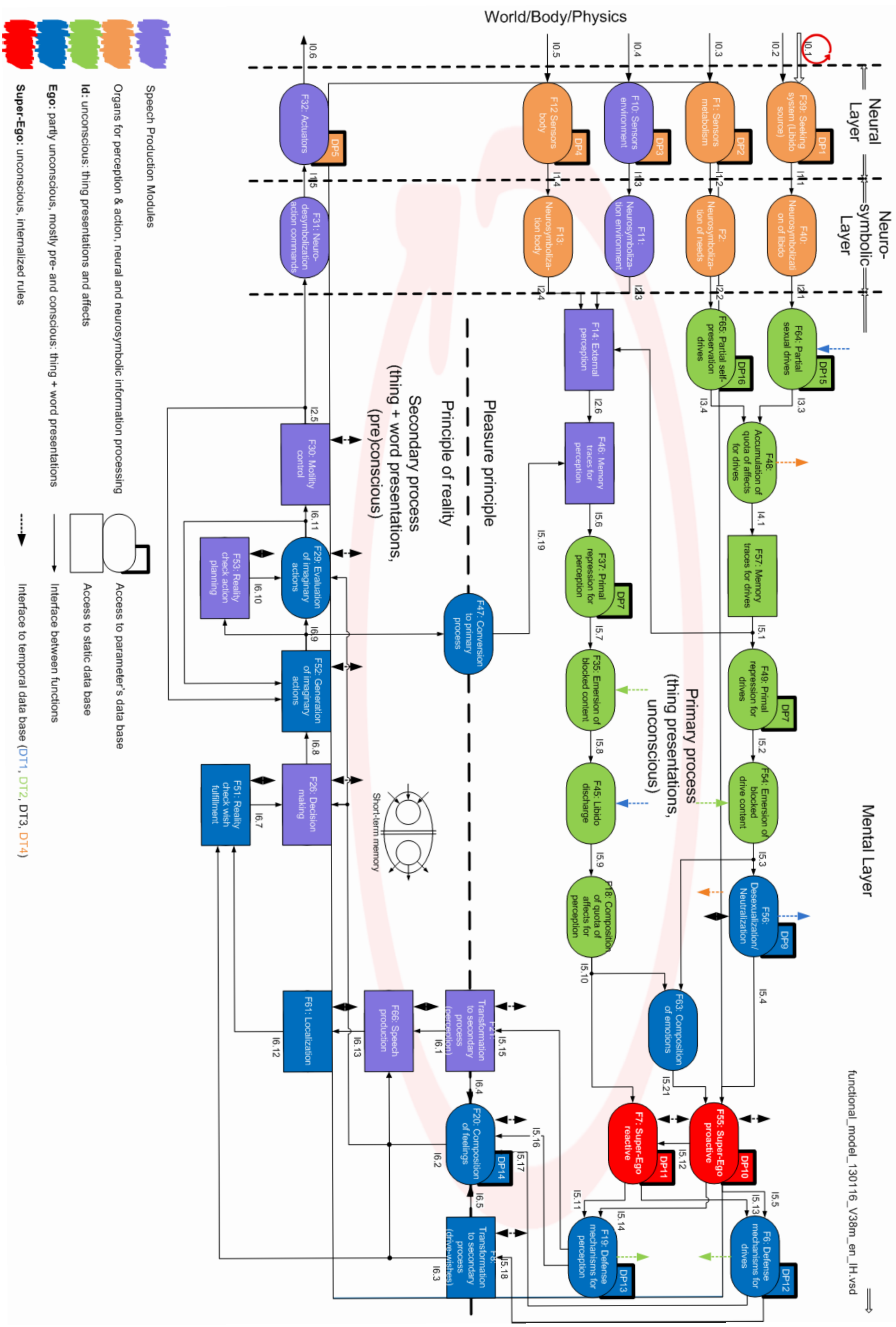

Figure 5.4: Simplified ARS Model Containing Language Modules

5.3 Realization of the Speech Production Model

The main goal of the following chapter is to explain how the underlying speech production model of Levelt is implemented in ARS model. To enable speech ability for communication, the integration of several functions that are part of the speech production process are required. These will be presented in the forthcoming subchapter.

First comes the central function of *Semantic and Syntactic Speech Ability*, which makes out the situation context and then generates a wording for the speech statement. The wording is the part of a WPM and specifies how to pronounce a word called loud image [10]. After the visual TP has been attached to the acoustic WP in F21 the selection of a context appropriate word can start. The word selection is not an unconscious process, but a conscious process that happens in the central module *Semantic and Syntactic Speech ability* (F66).

5.3.1 Semantic and Syntactic Speech Ability - F66

The central module F66 is responsible for the transformation of visual thing presentations to acoustic word presentations. An acoustic word presentation is needed in addition to the visual thing presentation (object presentation) that has been introduced by Freud in 1891 [Fre91] to generate meaning.

The module is required for retrieving inner and outer speech statements after a situational context has been recognized. The first step is however to recognize situational context. This is done based on the process that has been described in chapter three *Concept and Model*. In order to recognise a situational context, acoustic and visual perceptions are received, fusioned and translated to symbols in the primary process. In the secondary process they are grouped together as CEs. In addition, the module F66 needs access to the ontological knowledge base to access the wordings for the later speech statements. Wordings are also needed for inner speech, although it is not pronounces [LEHG12].

For each CE, the knowledge base contains a loud image of the wording that can be retrieved. The multiplicity is one CE for one wording which needs to be granted.

Furthermore it needs to be emphasized that in F66 the number of wordings are defined that can be uttered in a context. Thus, there is no definitive statement that is produced in F66 but a number of situation adequate statements. This is due to the fact that decision making finally decides which statement is uttered based on the goal list.

As for the generation of meaning, F66 is responsible for attaching a TP to a WP. The link between TP-WP establishes the meaning at this point. In the end a meaningful WP along with its acoustic loud image comes out of this module and can be processed further.

The module for *Semantic and Syntactic Speech Ability* F66 according to Freud is responsible for connecting the acoustic loud image to a thing presentation, which is exactly the step where the meaning is acquired. What happens is that all incoming thing presentations are matched with the categories supplied by the CEs. That is why the module entails semantics. The CEs represent a way of semantic matching according to known categories. It means that peceptions are assigned to semantic categories. The CE categories, that have been in introduced in chapter Concept

[10] A loud image contains the sounds how to pronounce a word.

(Action, Emotion, Distance, Entity and Drive) are filled with the according information from the word presentation meshes.

The retrieval of a statement is also realized in this module. In the domain ontology capturing situations the wording is stored with each statement that can be produced. This wording needs to be retreived in order to know how a statement is articulated, as the wording contains information abouthow to pronounce a word. The wording of each CE in the memory is sent in the next step to the following module in the ARS model.

The fig.5.5 below shows what happens in module F66. After the TPMs are received from the predecessing modules and the context has been recognized, the visual component of the TP is connected with the acoustic loud image. Based on this loud image, the appropriate concept for inner speech and outer speech is chosen.

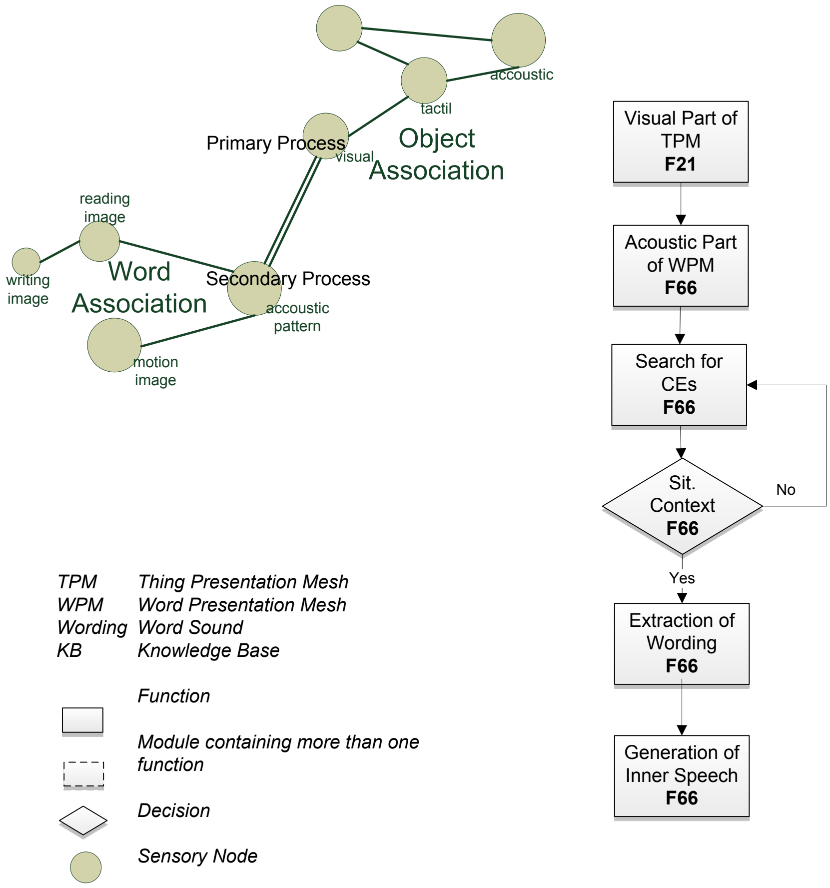

Figure 5.5: Connecting of the Thing Presentation to the Word Presentation

However, for the speech ability it is needed to recognize situational context, as such. Another point in addition is that a context is verfied by an reality check in F53. Also wish fullfillment can manipulate the production of a word after situational context has been found and the though has been selected. Plans about actions as well as plans on words can exist and thus influence the generation of a word.

5.3.2 Decision Making - F26 and Reality Check on Actions - F53

Another processing step that is required before execution is decision making (F26). This module generates a list of weighted goals from which depending on the weight the decision is made. The weighted goal which has the highest weight is send forward to the next functional module in the order. A goal list, where each goal is associated with a weight has the advantage that it can be searched by a greedy algorithm quite easily. The module is responsible for deciding, which goal is selected and which appropriate wording is produced in a certain context in F26.

Fig.5.6 shows the *Decision Making* module F26 that has calculated, which goal is selected. In between are the modules F52 and F29 and the next speech related function is F53, which is responsible for selecting the content that is said. The Outer speech statement generation depends moreover on the drive of how to say something. After the situational context has been established, the drive representative is taken into account and finally contributes essentially how to say something. For instance, if agent wants to greet someone in a situation there are two possible ways. Either he can say "Welcome!" or he can say "What?". In case the drive compound is libidinous, he will prefer "Welcome!" in contrary to "What?", because he anticipates the saying of a word. Anticipation means that the agent calculates the pleasure that his/her statement will cause. Thus, the anticipation depends on the pleasure level that has been introduced already.

Actually, it means the functional module *F53* retreives the amount of pleasure that can be achieved by an agent if a word is produced. The pleasure value is based on the wording and is stored together with each CE in the ontology. Thus, for each wording that will be produced the amount of pleasure is retreived from the ontology. In that way, F53 can still decide in between words, which one is uttered depending on the pleasure it will cause in a situation. The decision is in between several words that are appropriate in a situation. However, the word which has the highest pleasure level wil be chosen.

The next fig.5.6 shows only a small part of the whole ARS model. It presents the processing of the word presentation mesh plus wording from module F26 (Decision Making) to F52 (Generation of Imaginery Actions) then to F29 (Evaluation of Imaginery Actions) and last to F33 (Motility Control). In module F53 the WPM plus *wording* is compared with current situational context again. Psychic intensity in F53 is needed to retrieve the amount of pleasure achieved from a wording that is articulated. Psychic intensity is known as a synonym for psychic energy [DBM+13].

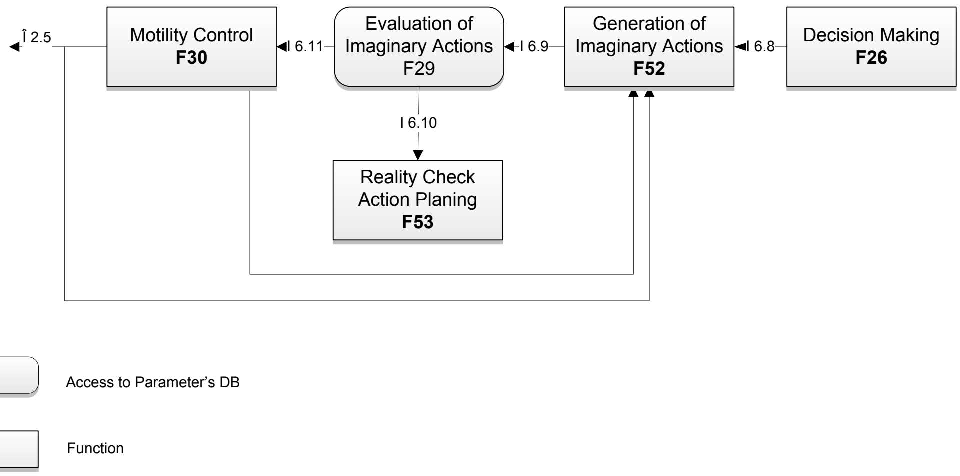

Figure 5.6: Modules connected to Decision Making and Reality Check

Hence, the decision, whether something is uttered as a statement is based on the amount of pleasure an agent can gain when producing a statement. Here the pleasure unpleasure principle kicks in by means of the pleasure value that has been explained in the upper section. In conclusion it can be said, the anticipation mechanism is based on the pleasure unpleasure principle.

5.3.3 Articulation Module and Motility Control - F30

The requirements for inner speech production is situational context and selection of the word for the inner speech statement. However, an additional module is required for the actual production of the speech statement. What is done here is that the word is articulated. From the already recognized context the concept entity is retrieved. In the case of thought production the context for psychic content is retrieved and given as an output finally.

The monitoring of Levelt can be implemented by making a feedback from the module F30 to the speech production module at F66. Monitoring prevents speech erros of any kind that occur in speech. In addition the monitor also checks inner speech for mistakes that can be made before speaking. The monitor thus covers the check of inner speech, whether they are misplaced at a certain situation. How does that work? The detection is as simple as the following. If the agent sees something he thinks about the things in his direct environment and the associations connected to that object. Thinking of a car, something completely different, when eating a cake is thus not an adequate thought in this situation that is detected. The importance for inner sppech in psychoanalysis is that patients complain they think about things they dont́ want to.

According to Levelt there is an inner and outer loop, which can be seen in the following fig.5.7 that is responsible for the monitoring of inner speech. However, in psychoanalysis it can happen that speech is uttered, where the monitor is less active. As a consequence speech statements can get to overt speech that are usually not intended for speech. This effect occurs if people are either very tired or if the monitor is not active anymore due to other attention reasons.

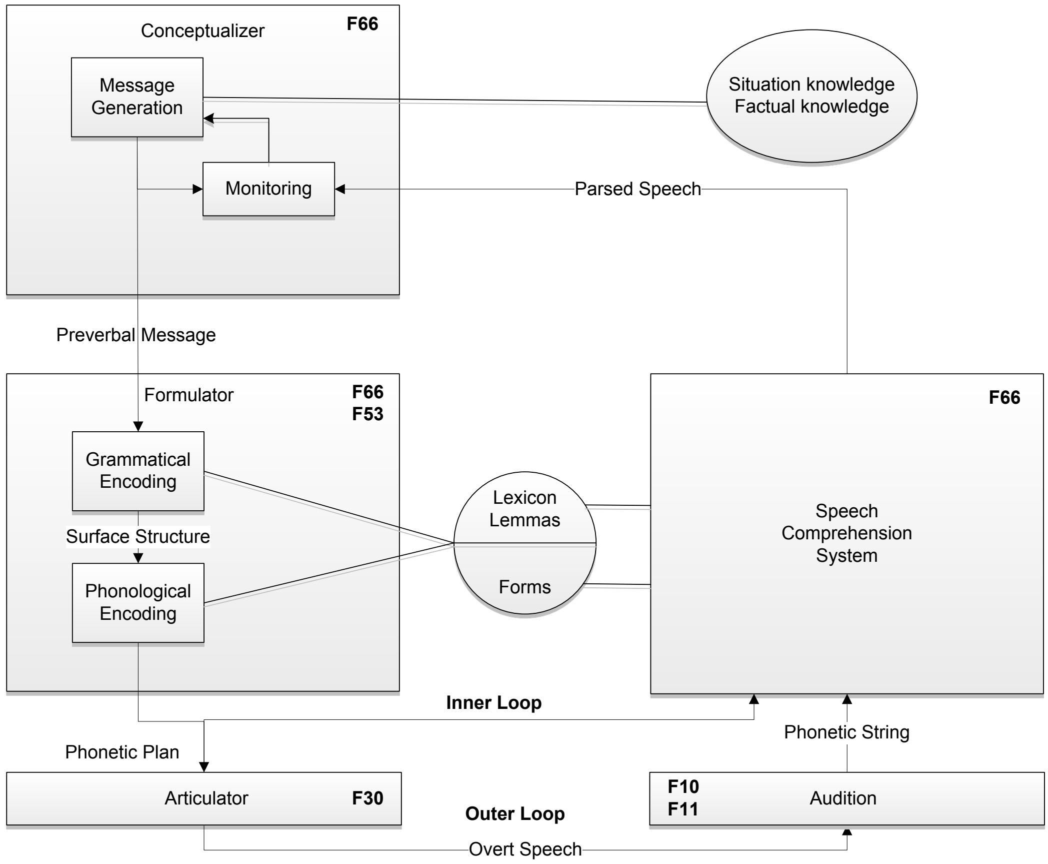

Figure 5.7: Monitoring in the Model of Levelt

As shown in fig.5.8 the inner loop is responsible for monitoring the word before it is uttered, whereas the outer loop is responsible for checking the word after it has been uttered. Both loops are intended for error correction. These are structures that were suggested by Levelt for error recognition.

Inner speech is not articulated, if the decision making module permits a passing, which is the case if no goal fits the equivalent outer speech statement. In this case the inner speech statement still is sent back for another iteration to F47 *Coversion to primary process*. It is what we call as thinking without speaking. The feedback loop can be seen in fact in the next fig.5.8 more clearly. The statment goes from module F47 to module F46 and is transformed yet to another TP plus acoustic content. The inner loop can be realized by feedback between the modules F47 and F46. After speech has been produced in F66 it is further transferred to F61, F26 and F53 for a reality check. However, if the reality principle detects an error than it will not be uttered and thus be sent back to F46 to undergo another cycle. In a second model cycle the statment is monitored and by coming in to F46 again the concept can be checked again. If the concept is changed the second model cycle starts and eventually the inner speech statement produced in F66 can be uttered by the motility control module in F30. The process is described in fig.5.8.

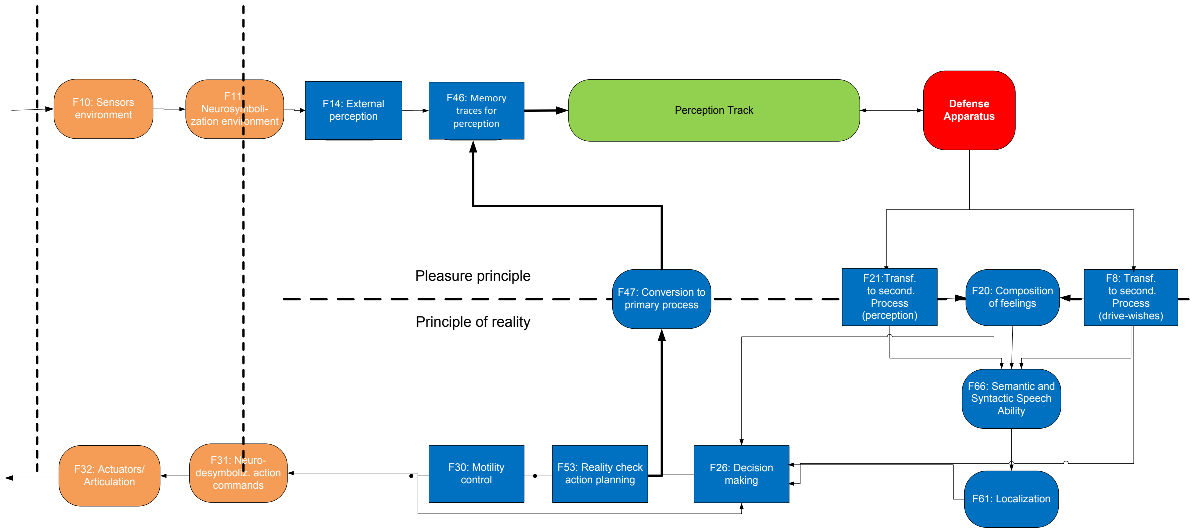

Figure 5.8: Feedback from Module F47 to F46

In addition, module F53 is needed for checking the reality in case the intended utterance does not fit with the situation. It depends on the reality principle to decide whether a statement is uttered or not. By speech comprehension it is actually meant, what happens when we hear a word and assign the object association to the acoustic wording. Last but not least, the articulator is entailed in the functional module F30.

5.3.4 Interfaces and Functions to other Modules

This part discusses all functions and modules involved with the new function of speech ability:

- F10: Sensor Environment ,
- F11: Neurosymbolization,
- F14: External Perception,
- F46: Memory Traces for Perception,
- F08: Tranformation to Secondary Process Drive,
- F20: Composition of Feelings,
- F21: Tranformation to Secondary Process Perception,
- F26: Decision Making,
- F53: Reality Check Action,
- F31: Neurodesymbolization, and
- F30: Motility Control.

To see how the functions are integrated in the entire ARS model an overview picture will be presented in fig.5.9 The overview shows all the speech functions that are integrated in the ARS model. In the very beginning the acoustic and visual input enters the model and is processed in F10, which resembles the sensoric senses. The module receives acoustic sensor data and transforms it to a value that can be used in the further mental apparatus. Module F11 allows the neurosymbolization of the sensor data, meaning that sensor data is transformed into a multimodal neurosymbol. In case of the acoustic sensor data it is transformed with other sensor data into a multimodal symbol. Then module F14 transforms the neurosymbols into TPs. The TPs are associated by means of their spatial-temporal relationship and with the TPs. As a result TP meshes are achieved. The acoustic data that is part of a neurosymbol and is associated with the TP in a spatial-temporal way. In module F46 all entities are matched with the perceived image (PI) and in a next step memories of these are activated. In addition, the module contains long-term storage and the self entity. The first component is needed for accessing memory traces from the past. The latter one is needed for binding the body perception to a self object. In the preceding modules the input is processed accordingly until the defense apparatus, which lets the TPMs pass that are allowed to enter consciousness. A further module that is needed for the speech ability is F08 *Composition of Feelings*, where the emotional component of the function is needed for the formation of the CEs. Apart from this, from F20 *Tranformation to Secondary Process Drive*. This module offers the drive component and forwards it to the novel *Speech Ability* module, where the CEs are formed. The third component that is needed for the CEs is the visual TPM that is associated with the acoustic part of the WP. Thus, module F21 *Tranformation to Secondary Process Perception* connects the visual part of the TPM which is the actual object itself with the acoustic part of the WP in F66 *Speech Ability*. The central module F66 will be described in the upcoming chapter. However, the next module, where speech is taken into account is the decision making in F26. It is responsible for deciding which of the speech statements or wordings are situation adequate and should be uttered. After the decision has been made the next module where the processing goes to is *Reality Check Action Planning* F53 where the reality check of action planning is made. This module checks in accordance with the reality, whether it pays of to say something actually. Before the statement is articulated the action needs to be neurodesymbolized in F31, meaning the neurosymbol is translated to an actuator action again. Finally the motility control in F30 takes over and the inner speech is actually uttered as a statement.

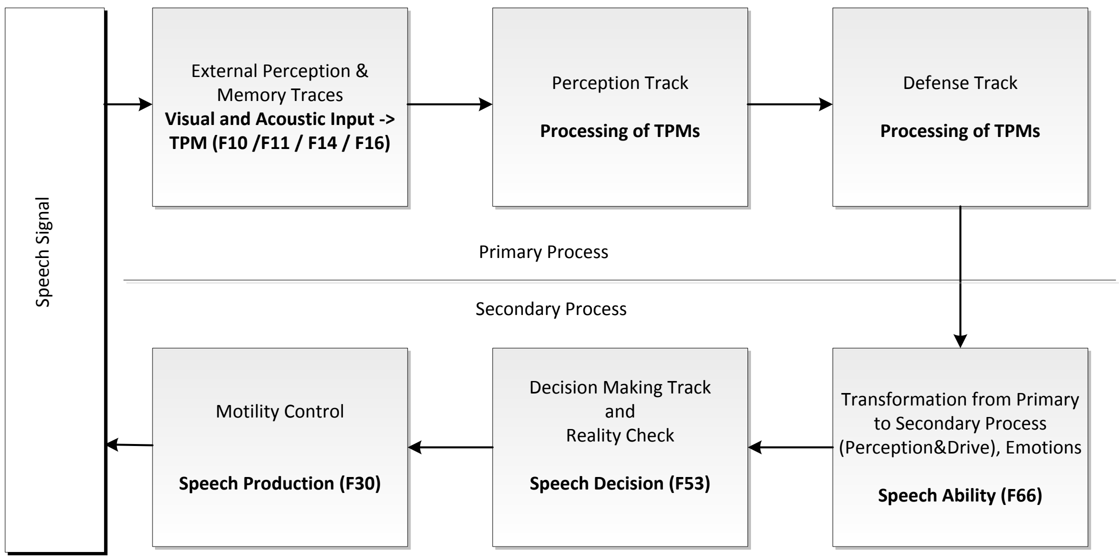

Figure 5.9: Overview Speech Ability Functions in the ARS Model

5.4 Realization of Listening as an Reaction to Speech

After the agent says something the other agent can receive the statement if he is nearby. Actually the agent has to have the speech statement in the radar of his own. There is no distinction made in whether the statement is far or near. The speech statement has to be only in the acoustic radar of the agent. The acoustic radar in fact resembles to the visual radar.

Listening is defined as the reaction that the other agent has when he receives a speech statement and processes the statement.

The processing of speech during listening goes on the same pathway as if speech is generated for the first time. Humans listen to speech by using their ears as receivers. Then speech is processed in the auditory system further. In the simulator however, speech enters at F10, where it is recognized as acoustic input with an auditory flag. Thus, the agent knows that he has not produced the statement. In F11 the neurosymbol is created and sent to F14 where they are transformed to TPs. In F46 memory traces are associated to the TPs and the acoustic content and then the processing is the same until the acoustic content reaches F66. In F66 the context is obtained and the received speech statement is categorized according to the context. Thus, if the agent Bodo has said "Share?" the reaction of Adam is situation adequate. Hence, Adam is not going to flee or run away if Bodo offers him to eat the cake together. Else the situational context has not been recognized correctly and the agent reacts inadequate or out of context.

Fig.5.10 shows the acoustic radar, where the statement has to be within the center radar to process the statement accordingly.

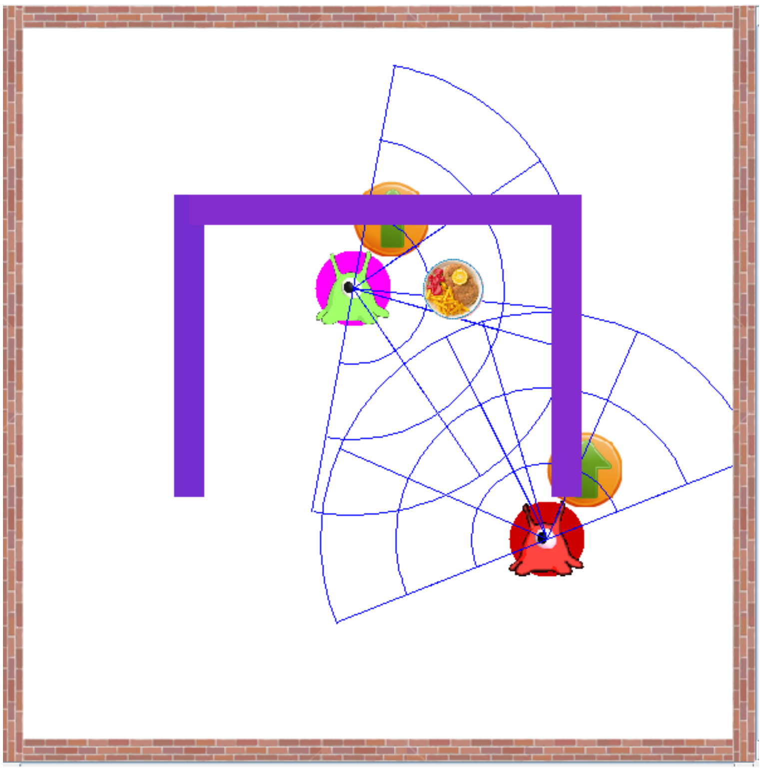

Figure 5.10: Acustic Radar

The fig.5.11 below visualizes how a speech statement is processed in the ARS functional modules. As it can be seen at first the acoustic perception is processed in F11 and F12 before the wording attached to the TPM is sent to the primary process. In F46, which is in the primary process module, the memory searches for the appropriate TPM that is associated with the wording (acoustic perception) that was heard. For example, if one has heard the statement "Yes", the TPM for "Yes" is activated in the primary process and it's contextual entity CE: Object is activated in the secondary prcocess. In F21 the transformation occurs from the primary to the secondary process. In F66 the process includes the heard word in the form of a CE. Yes is transformed into a CE: Object when it comes from the TPM in F21. Thus, the heard statement contributes to the situational context.

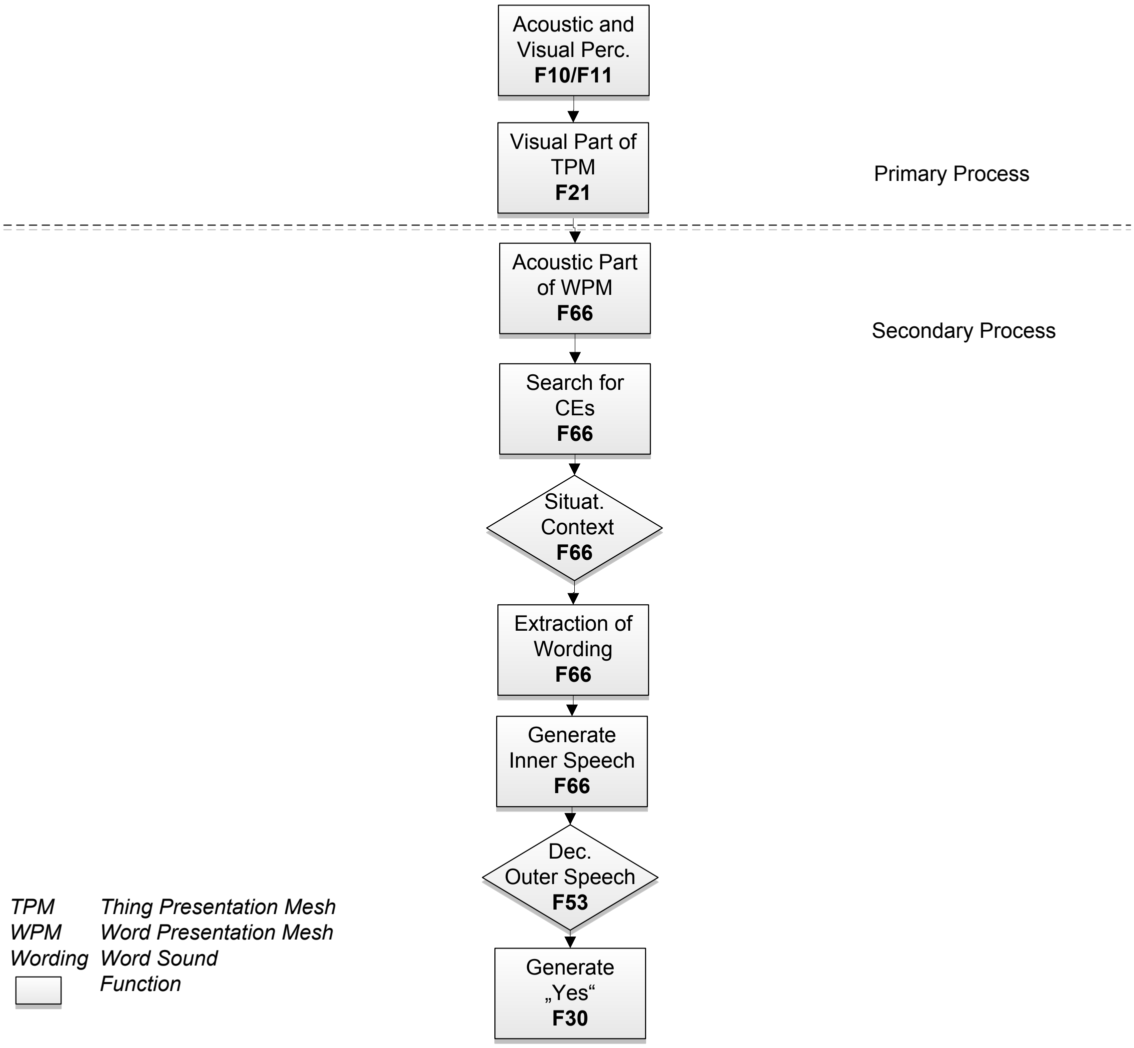

Figure 5.11: Overview Statement Generation of "Yes" in the Process Chain

5.5 Realization of Alternative Actions

The following chapter describes how actions are selected and executed from a list of goals. The underlying mechanisms therefore are described.

In the current implementation at hand based on Wendt [AWMB13] goals are generated and out of a goal list the goals are selection for a coming action. The selected goal leading to an action is derived from the situational context as well. In the implementation ARSi13, for alternative actions the list of goals is derived from the situational context. From there the three best matching goals are taken into account for action generation. For example, if the list of goals contains: tea drinking, coffee drinking and beer drinking and the context is set by the "context: Eat Schnitzel". Then the best matching goal is beer and tee drinking, because in the context for eating a Schnitzel, it is more appropriate to offer beer or tee. Coffee would be a more appropriate action for, if they would eat cake.

In summary what is new in the implementation, is that the goalList is compared with the situational context. The list of goals needs the acquired context as a minimum for generating an action. As regards the functional location of this method, the generation of context takes place in module F66 and the generation of goals takes place in module F56.

An implementation of action selection is based on a list of goals taken and the action that depends on the CEs in the situational context. For the execution of actions a selection meachanism is needed. Meaning, a match function determines which CE out of the goal list fits the context. For example, "CE: Eat" from the goal list fits better to the "context: Party". Thus, the goal list is searched through to find a CE that fits the situational context. Then the CE is transformed to the according "CE: action" in F30. For "CE: Eat" the action execution is "Eat" then.

5.6 Realization of the Use Cases

The section describes how the use cases as described in the Technical Review [DBM+13, p.80] are implemented to evaluate content. Use cases serve in order to see if the requirements for a specific psychic content have been solved accordingly. In the work at hand the use case one is supported by inner speech and outer speech, whereas use case two, which is extended for actions.

5.6.1 Realization of Use Case 1 extended by Inner Speech and Outer Speech

The subchapter is based on the use case one from chapter *Concept* that has been adapted to serve the purpose of inner speech and outer speech. In principle, the first use case has been taken as an example. It has been modified in the way that speech and thought are added in a certain context.

Clarification of the Concept

The scene starts with Adam in the house and Bodo being in the near sourrounding. Apart from Bodo, who owns the house a food source is nearby, but Adam is not aware of the food yet. The scene is adapted to use case one in that way the objects and certain background knowledge is added, such as the knowledge of a house that is owned by Bodo. Bodo is a successful person and is not very fond of Adam. The reason for adaption is a unique context is needed for producing speech that is triggered by situational context. The major modification of use case one is the extension of thought, speech and an action. What happens after a first scan is that Adam in case he is hungry generates the thought statement "Nourish". In a next step Bodo produces the statement "Pleasure", after he has seen the food source as well. However, Bodo is not really hungry. However, when Adam notices Bodo he asks, whether he wants to share the food source by using the appropriate statement:"Share?".

Regarding the content the use case is extended by the following mechanisms:

- Recognition of Context,
- Thought Statement Generation, and
- Speech Statement Generation.

5.6.2 Realization of Use Case 2 extended by Listening

The second use case is extended for speech and thought and action, whereby action taking is a further step.

Clarification of the Concept

The use case 2 is split into three parts due to modularization of the different contextual situations in this work.

The first part of the second use case starts with Bodo in the house, Adam entering and the food source belonging to Bodo. The situational context is set when Bodo is reminded of Bella by looking at the cake. However, the situational context changes dramatic when Adam enters the scene and goes for the cake. As a dialog is started where Bodo asks Adam:“Invited?” Adam responds with an appropriate action like going towards the cake and Bodo as well by waiting.

In the second part of this use case Bella appears in the scene for visiting Bodo. The situational context is Bodo recognizing Bell and the food source. She comes into the scene thinking “Bodo”. When she enters Bodo has the thought presentation "Bella" and Adam thinks about the food source that he wants to eat. Bodo welcomes Bella by saying "Welcome" and Bella’s action is to look at Bodo and Adam and ask him whether he is invited.

The third part of the second use case deals with the fight of Bodo and Adam. Adam looks at Bella and thinks "affair", while Bella thinks "calm" and Bodo thinks about "no affair". In the second step the dialog kicks in and Adam verbally attacks Bella by saying something rude, whereas Bella asks Adam if he is "Inivited". Bodo attacks Adam now by asking him to fight.

The next action can be one out of three:

- Bodo listens to Adam saying something,
- Bodo listens to Bella saying something, and
- Bodo listens to himself saying something.

The implementation at hand covers the following:

- Recognition of Situational Context,
- Inner Speech Statement,
- Outer Speech Statement, and
- Listening to a Speech Statement.

6 Simulation Results and Discussion

Verba docent, exempla trahunt! Translated from latin: Worte belehren, Beispiele reißen mit!

The following chapter comprises a summary of the simulation and simulation results for the thought and speech statement. The chapter also includes an overview of the used overlay icons in the simulation as they are part of the simulation. Based on the simulation results an overview of a customization of the speech and thought statement will be given for the reason that speech and thought statements need to be customized according to different use cases. Added to that, a discussion on simulation aspect covers those points that were solved during the implementation and simulation phase. Finally, the results of the work are compared to existing models that have been presented in the chapter state of the art.

6.1 Simulation and Simulation Results

The simulation results are broken up into the two existing use cases. The use case one of Adam looking for food will be described at the very beginning showing agent Bodo and Adam in the same location. Depending on the recognition of the context different thoughts and speech actions may occur. The same applies for the second use case, where the scene is extended with another agent Bella. In general, the simulation results are visualized by means of a graphical representation of blocks, showing the agent's thought presentations from module F21. Finally, the chapter simulation results includes a description of the overlay action icons that will be shown during a simulation. They are needed for visualizing the speech and thought statements as well as several other statements.

6.1.1 Outline Overlay Icons

Before the simulation results are explained thourougly an outline of the overlay action icons is given. They are used in general to show, what the agent is doing. That way it can be seen as an additonal opportunity apart from the sensors. In the simulator environment there is a separation in between motion icons and action icons.

Motion Icons

Motion icons are used if the agent is moving up and down, or left to right. As it can be seen in the figures below they are visualized followingly:

Figure 6.1: Motion icons for up and down, left and right

Action Icons

Action icons are used for indicating an action that is conducted by the agent. For example, when the agent eats, sleeps.

Figure 6.2: Action Icons for Eating and Sleeping

Outer Speech and Inner Speech Icons

Finally, the overlay action icons for thinking are presented. The so-called thought icons include the content extracted from the objects or persons, such as Nourish, Relax, Schnitzel, Adam, Bodo and furthermore. For that reason only four icons are shown here.

Figure 6.3: Inner Speech Icons

The speech icon contains one word sentences. They are for example, statements, such as Eat, Share, Affair, Attack.

Figure 6.4: Exemplary Outer Speech Icon

The information in the inner speech and outer speech statements is represented by a figure containing letters and can be customized by changing the content of the figure.

The use case in general is selected by means of the scenario selector that is part of the setup of the MASON simulation environment. The GUI of the scenario selector is shown in fig.6.5. It is a selection interface programmed in Java. The selection is based on the profile that is selected in the background. After a selection has been done in the GUI the profile is activated and the situation is loaded.

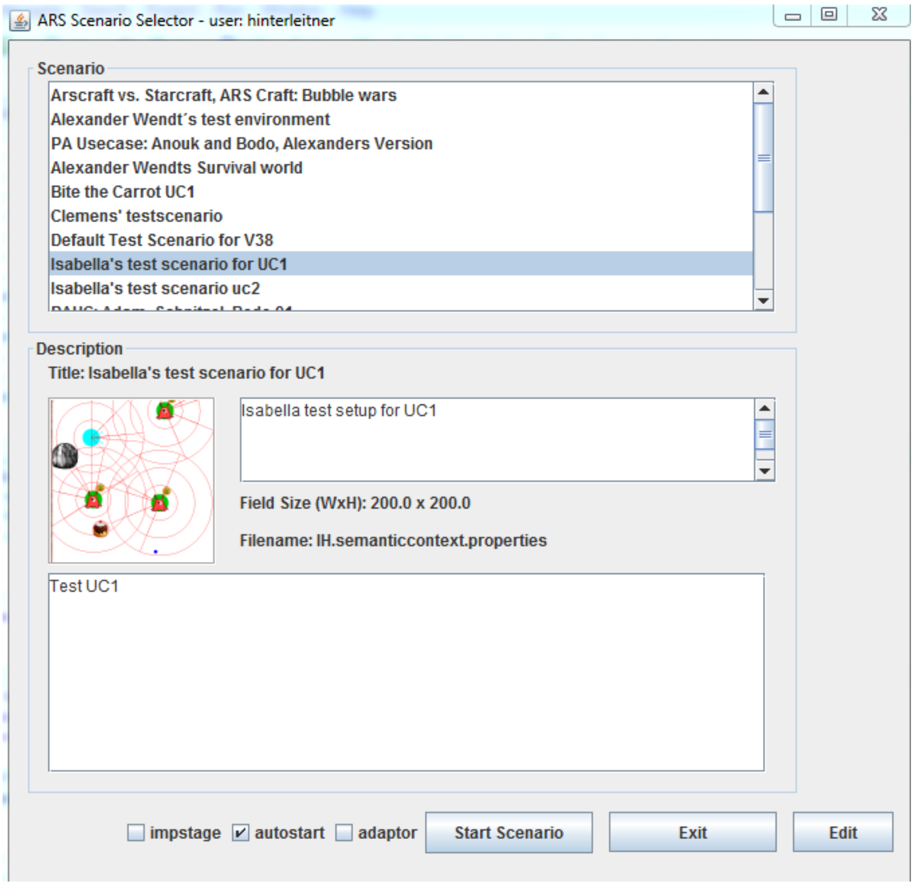

Figure 6.5: Scenario Selector

6.1.2 Use Case: Situational Context

To recognize a situational context as already developed and described in Chapter *Concept*, the perceptional input of the acoustic or visual type is processed. To understand how the process of finding situational context is realized fig.6.6 is presented. The figure presents how situational context is recognized based on the five categories and how it is linked to the functional module F66.

The incoming perceptions are ordered according to contextual entities that are categories, such as action, distance, entity, drive and emotion. If all five CEs match with the memory stored in the knowledge base then situational context is recognized and the wording can be retrieved. The match needs to be a full match of all five CEs, meaning every category contains the information of a CE. The left part of the fig.6.6 shows how the mechanism of situation recognition works, whereas the right part of the figure shows the processes in the module F66.

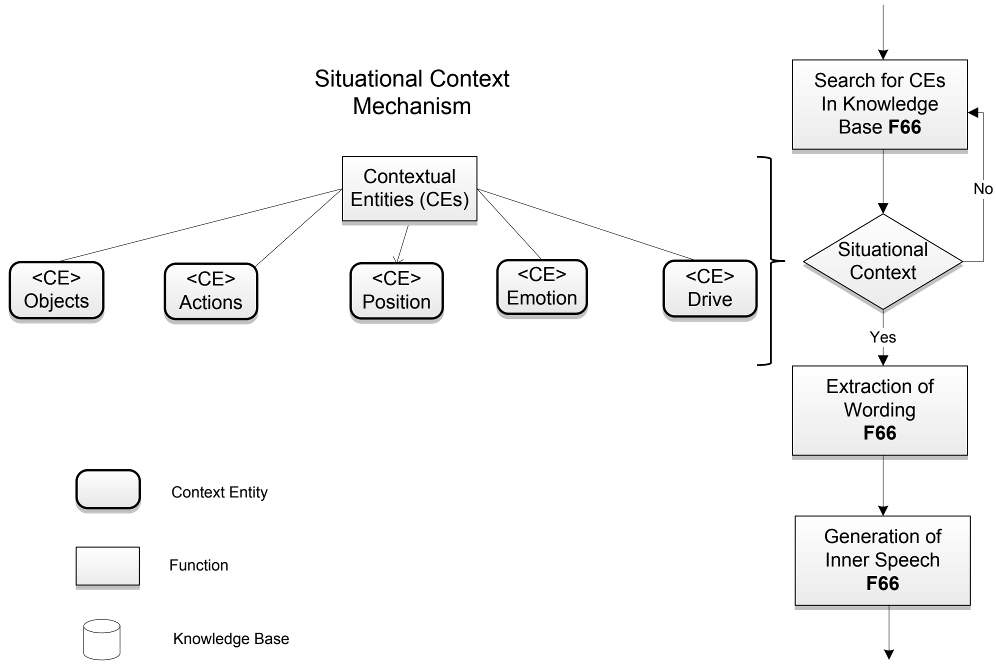

Figure 6.6: Recognition of Situational Context

The matching process is a limited by the amount of categories selected for identifying a situation. The finding of the CEs is a performance issue, because the look up of the CEs can be time intensive. In order to find the matching CEs and achieve a better performance, there is an option that a weight is attached to the CEs in the knowledge base. The weight indicates what impact a certain CE provides to a situation. Another option is that more than one CE is valid for a category. Both options will be explained in greater detail in this chapter in section *Discussion on constraint-based algorithm*.

Finally, the possible situational context configurations that can be recognized from memory are presented in the table below. The inner speech perspective in the simulator is visualized by the inspectors for inner speech in addition.

The result of the process in fig.6.6 is a speech statement based on the situational context. Table ?? assigns the situational context to the according inner speech and outer speech statement. The result is a matching table that sets a certain situational context for a statement. If the context "Welcome" is recognized the equivalent outer speech statement is "Welcome" or "Hi". In case something is missing from the inner speech statements (middle row), the simulation waits a cycle to update the missing statements.

Situational Context	Inner Speech	Outer Speech
Context Welcome	Bella, Agent, House, Near Bella, Move, Joy, Relax	Welcome! or Hi!
Cont. Ask-Listen	Bella, Agent, House, Near Bella, Move, Nourish	Others?
Cont. Ask-Listen	Bella, Agent, House, Near Bella, Stand, Joy, Aggressive	Known?
Cont. Listen-Answer	Adam, Agent, House, Near Adam, Stand, Joy, Relax	Yes.

Table 6.1: Semantic Context Situations Based on Use Case Two

6.1.3 Use Case: Inner Speech

Use case two that has been described in Chapter *Concept* will now be illustrated based on the results for the inner speech statement at first. Fig.6.7 shows the first picture of the second use case. Adam and Bodo nearby a Schnitzel in a house and the inner speech statement generation. Inner speech information is represented by means of the thought bubble in general. More detailed information on inner speech can be seen via the inspector tool, which will be described in the subchapter *Simulation Inspectors for Inner Speech*.

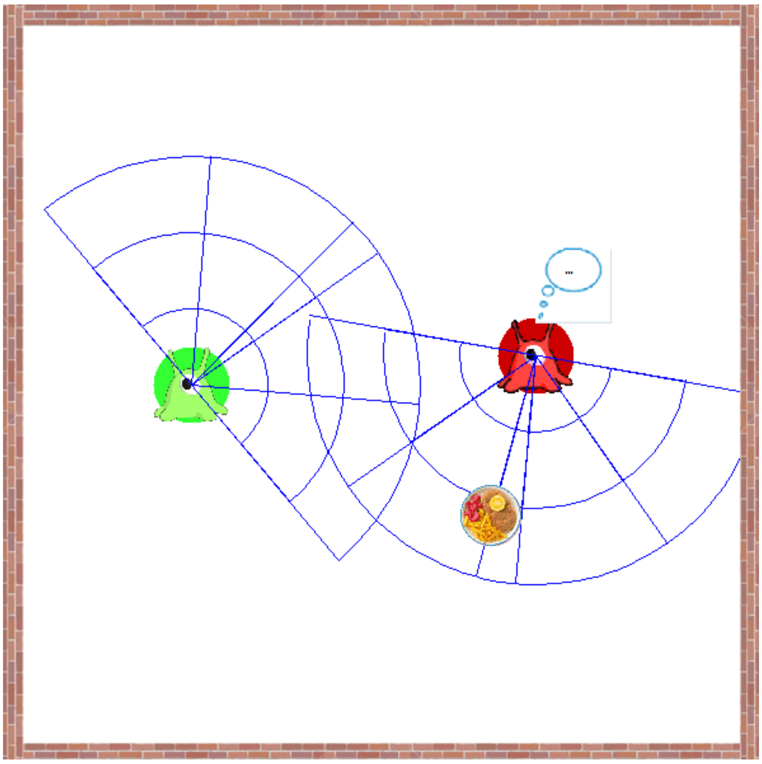

Figure 6.7: Use Case Two - Inner Speech Visualized by Three Dots

In order to produce inner speech, such as "Schnitzel" and "Bodo", Adam has to recognize the situational context before. As it can be seen in fig.6.8 the process starts with the incoming acoustic and visual perceptions that are processed until the situational context search is done. If the SC search matches all CEs successfull, then SC is recognized and the wording is extracted. Depending on the CEs, the corresponding wordings are retrieved and the inner speech statement is formed.

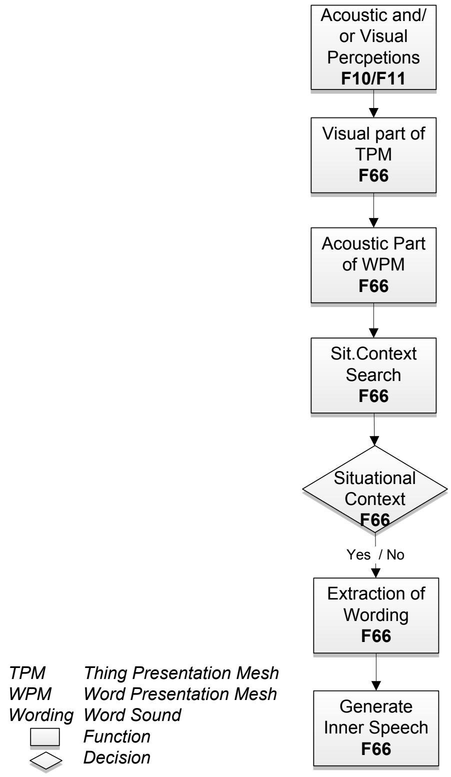

Figure 6.8: Process of Inner Speech

Simulation Inspectors for Inner Speech

To see what kind of inner speech is formed in a certain situation, it is needed to look at the inspector tool in the simulator. This can be done by selecting the agent, pressing the button *Brain Engine* and going to *ARSIN Overview and Concepts*.

- CE Memory
- CE Drive
- Situational Context
- Wordings

The inspectors are explained in the following list:

Contextual Entity Memory: The CE memory can be seen as a buffer for CEs, meaning thoughts the agent has within a certain amount of cycles. This buffer lasts for a number of cycles that can be set. The last item in the buffer resembles to the outer speech statement that will be articulated.
Contextual Entity Emotion The reason why the *CE: Drive* is externally represented is, because it is retrieved separately. The drive is needed for triggering outer speech. The drive component,

whether it is libidinous or aggressive plays a major role for how one says something and can be seen in addition to the original drive.
Situational Context When context is detected after comparison with the knowledge base it will displayed in this inspector value.
Wordings The wording is retrieved if the SC is set. In this inspector one can see the wording after the context has been detected.

6.1.4 Use Case: Outer Speech

Within Use Case 2, outer speech is produced in a certain situation. That is why, the following fig.6.9 describes the initial situation of this use case. Adam is in Bodo's house. It is not his house and he is looking for a place to write when Bodo enters the house. He greets the new person in respect to the fact that he thinks Adam is inivited to the party as well. Bella is near to the house but has not entered the situation.

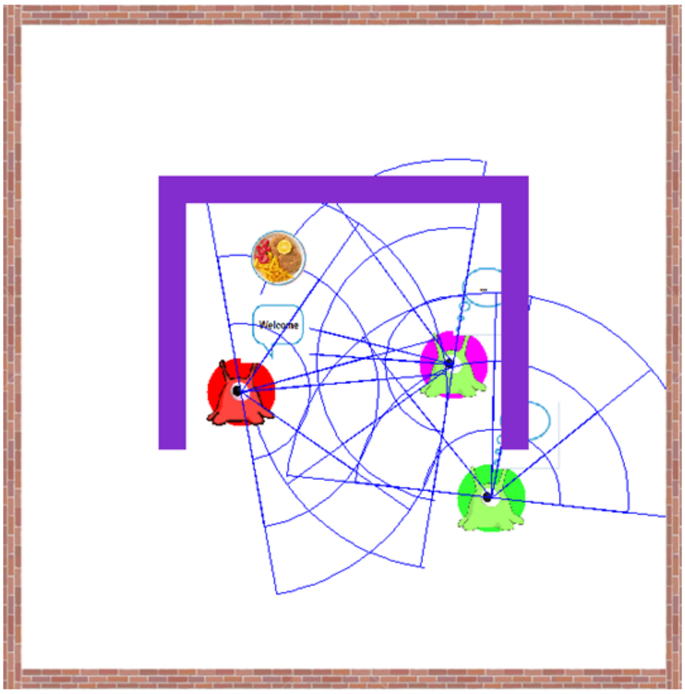

Figure 6.9: Situational Context Outer Speech

In addition, outer speech builds upon inner speech as it can be seen in the example 6.10. The fig.6.10 shows the process steps in functional module F66. At first five CEs set the situational context. After the context is recognized, the wording "Welcome" is retrieved for the situational context. Then the decision for outer speech is done depending on the drive value. If it is positive the value welcome is uttered in the end.

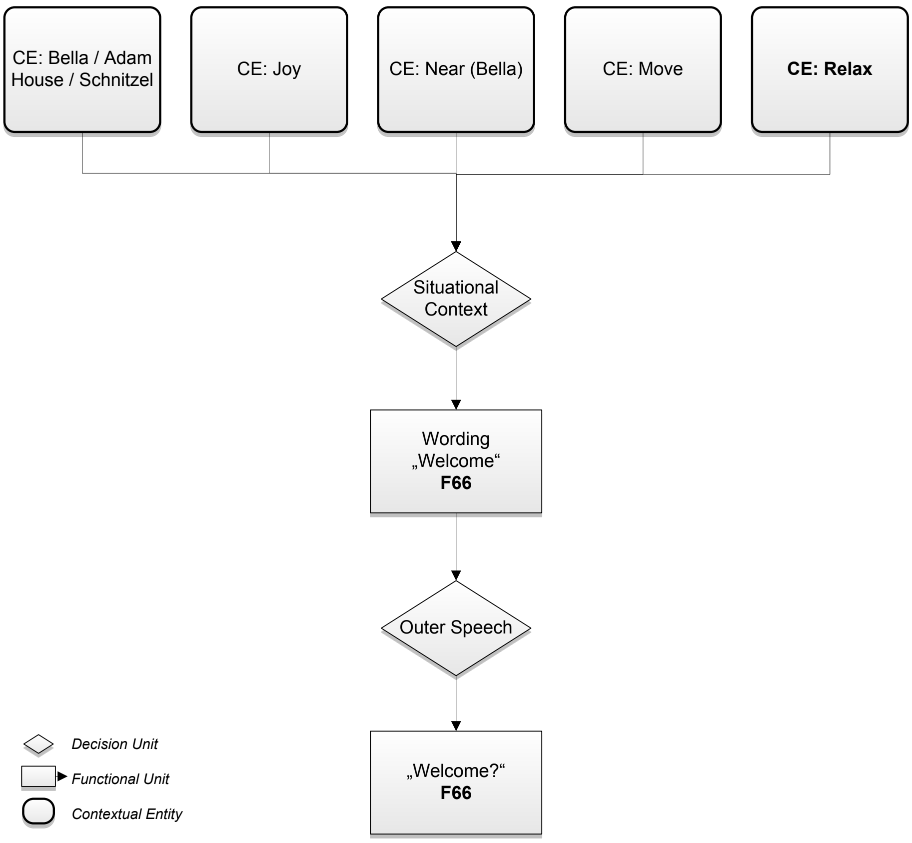

Figure 6.10: Speech from Bodo's Perspective

Simulation Inspectors for Outer Speech

Outer speech can be observed in the main window of the simulator as seen in seen in fig.6.9. Inspectors for outer speech are needed to view the results of fig.6.10 and fig.6.12. In addition, the inspectors of the speech production module (F66) enable to inspect the outer speech processes and how they are built from inner speech.

Figure 6.11 shows the simulation inspector for outer speech, which is called articulatory WP in the inspector frame. The reason for this expression is that the outer speech statement once it has been uttered is an articulatory speech statement. The articulated WP in the figure is "Share?" whereas the auditive WP is none, because the agent is only articulating and not hearing something. In case the agent hears something the auditive WP would also contain a value.

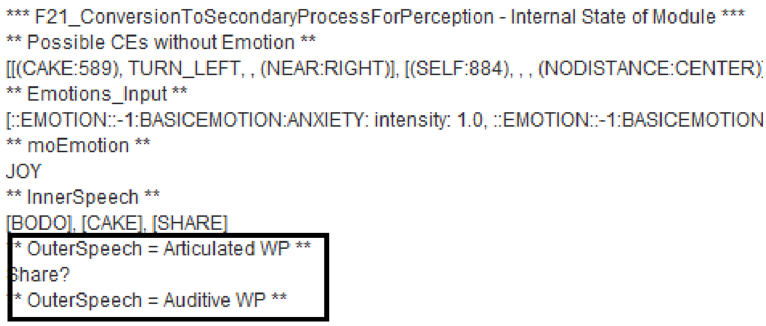
*** F21_ConversionToSecondaryProcessForPerception - Internal State of Module ***
** Possible CEs without Emotion **
[[(CAKE:589), TURN_LEFT, , (NEAR:RIGHT)], [(SELF:884), , , (NODISTANCE:CENTER)
** Emotions_Input **
[::EMOTION::-1:BASICEMOTION:ANXIETY: intensity: 1.0, ::EMOTION::-1:BASICEMOTION
** moEmotion **
JOY
** InnerSpeech **
[BODO], [CAKE], [SHARE]
* OuterSpeech = Articulated WP **
Share?
* OuterSpeech = Auditive WP **

Figure 6.11: Inspectors for Outer Speech

6.1.5 Use Case: Listening to an Agent

Fig.6.12 shows the process of listening, where an **articulatory or auditive** statement is heard. The processing of acoustic input does not differ much to visual input. It is stored within a TPM and transported to the module F21, where the visual part of the TPM is attached to the auditive part of the WPM. After the situational context has been detected for a heard speech statement it is interpreted according to a semantic assignment.

Thus, after hearing the agent say something, an reaction occurs in the form of inner speech or outer, articulated speech.

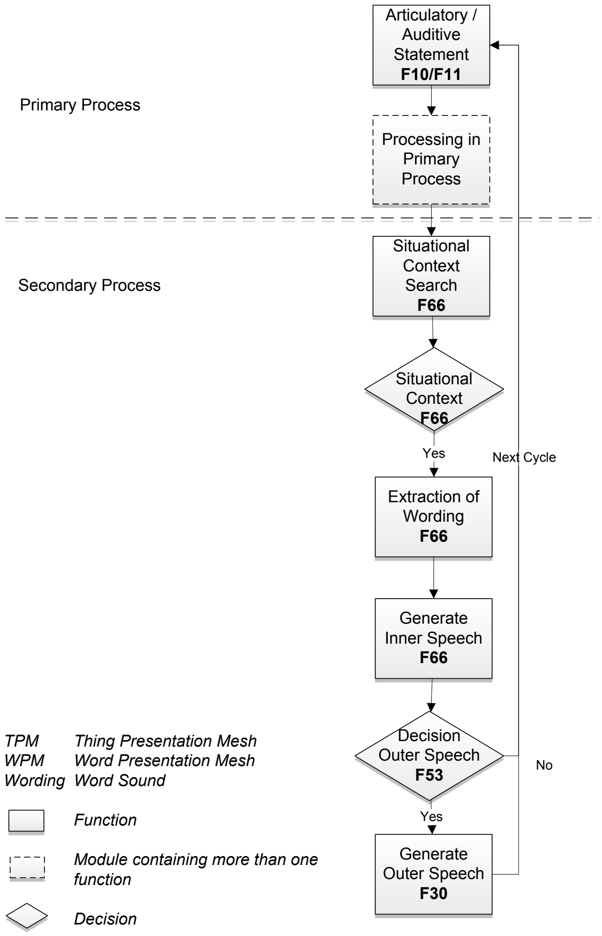

Figure 6.12: Process Listening to an Agent

The example of fig.6.12 focuses on the listening process based on the second use case and shows the intital requirements for listening. In the situation the agent has to be near the other agent and receive the speech statement as an auditive statement in the acoustic radar as it has been explained in Chapter *Implementation*.

6.2 Situational Context: Ask and Listening

Based on the table in the beginning of this subchapter *Ask and Listening* the results will be illustrated. Regarding use case two from the Technical Report [DBM+13], it will be demonstrated what is required to achieve the context Welcome. Use case two, situation 2 is introduced here, in order to demonstrate here, how the asking and listening process works. In fig.6.13 the CEs for setting the situational context is shown. The CE Entities: Adam and Schnitzel, the CE Action: move forward and the CE Distance: Near Bella as well as the CE Emotion: Joy and the CE Drive: Relax are represented. The next step after situational context recognition shows the possible CEs including the wording. Last step is the final generation of the retrieved wording.

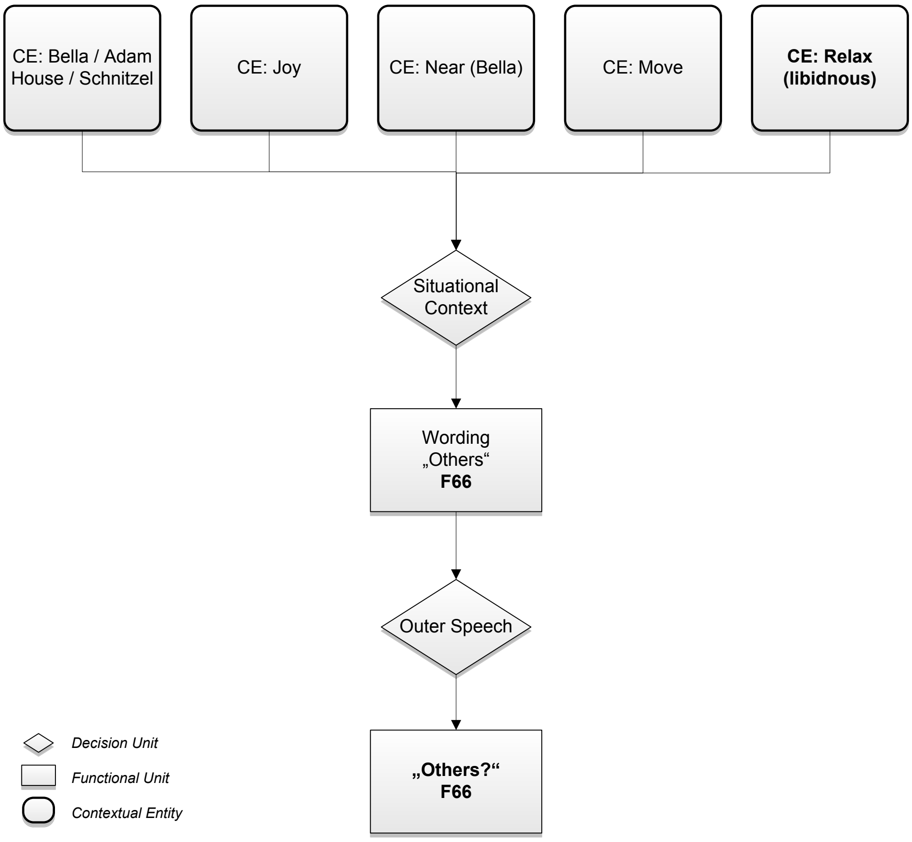

Figure 6.13: Situational Context Others

Fig.6.14 shows the Protégé perspective of the process described in flowchart 6.13. As one can see the the *CE: Drive* and *CE: Distance* are a precondition, meaning that if they appear in that context they are a precondition for the consequence *CE: Action*. The *CE: Entity* section contains all the objects and persons in the situation. *CE: Action* contains the action "move forward". If the consequence takes place the wording "Welcome" of type string can be retrieved. For the planing module the anticipation factor, which can be set to either 0 or 1, is needed.

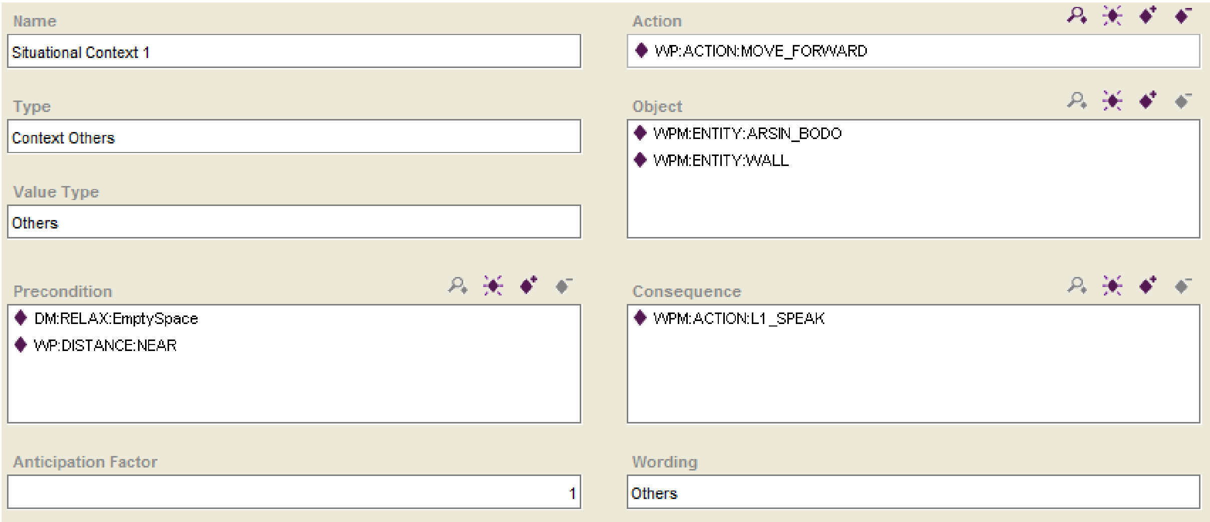

Figure 6.14: Protégé Representation for Situational Context Others

In fig.6.15 the ability to listen to a speech statement is demonstrated. In comparison to the situational context *Situational Context Others*, this context frame entails the people Bodo and Adam, the house and a schnitzel, the emotion joy, the action stand, and the location near Bella. It is seen from Adam's perspective. The whole process takes place in the module F66 as well. In difference to the process generation of outer speech, the listening process takes into account the last said statement "Known?" in combination with the whole context. When the articulated statement "Known" and the context have been recognized in module F66, the answer "No" is formed based on the context and the question.

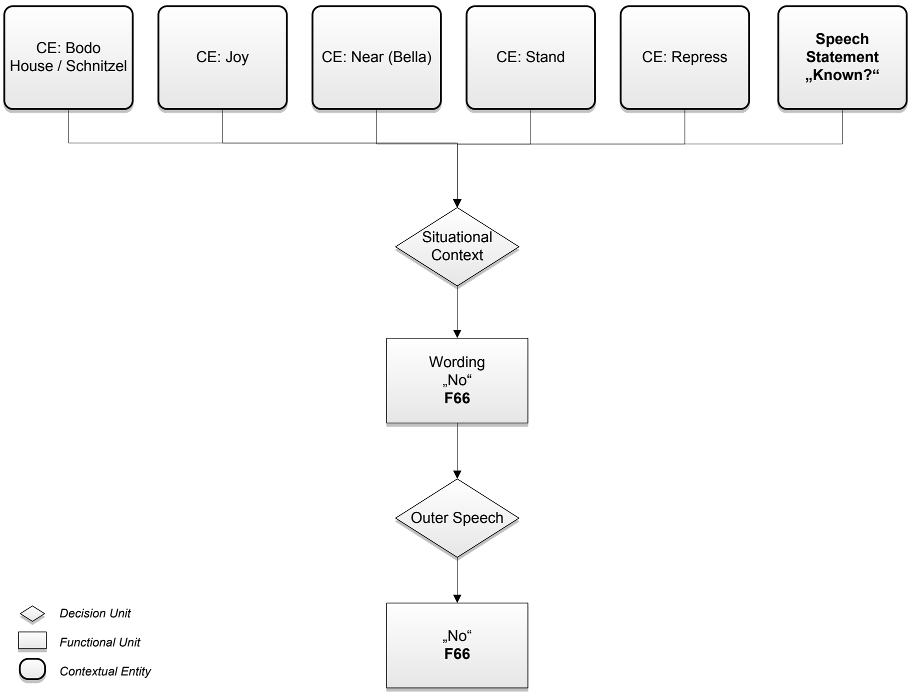

Figure 6.15: Situational Context Integrating the Statement Yes

The listening statement is received, if the agent is nearby the other agent. Meaning, to listen to a speech statement, the statement needs to be in the red area of the acoustic radar as it can be seen in fig.6.16. In the case shown below the agent can not hear, what is uttered by the other agent.

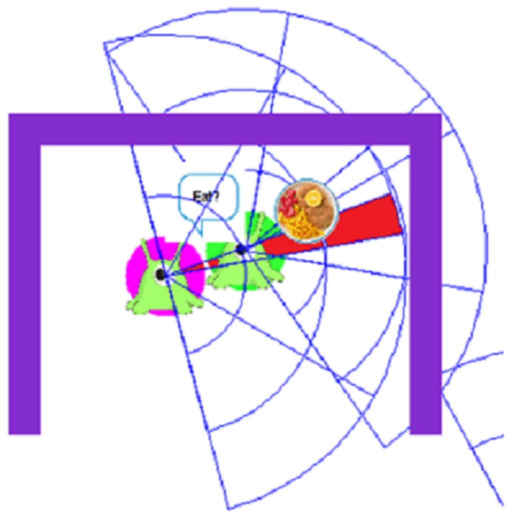

Figure 6.16: Acoustic Radar

The fig.6.17 shows two windows. The main window (sourrounded by the wall) and the inspector window below. In the main window one can see Bella in the house and Bodos near the house.

In the second window below the inspectors for Bella can be seen. The articulated WP "Eat?" is uttered by Bella and the Auditive WP is empty, because nothing is heard.

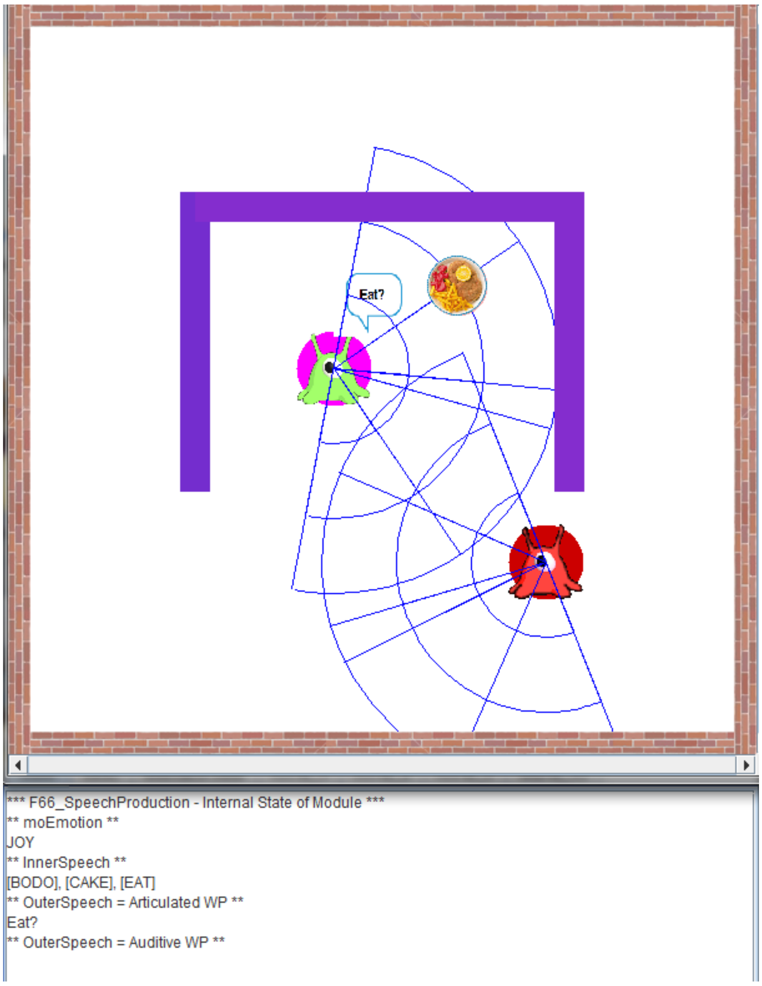

Figure 6.17: Overview Speech Statement

Simulation Inspectors for Listening

The statements received can be seen in the inspector tab as well as presented in the fig.6.18. Between listening statements one distinguishes between an auditive WP and articulatory WP. As explained in chapter *Implementation* the articulatory WP is used as a flag, if an agent produces an utterance. The auditive WP on the other hand describes if an agent receives a speech input.

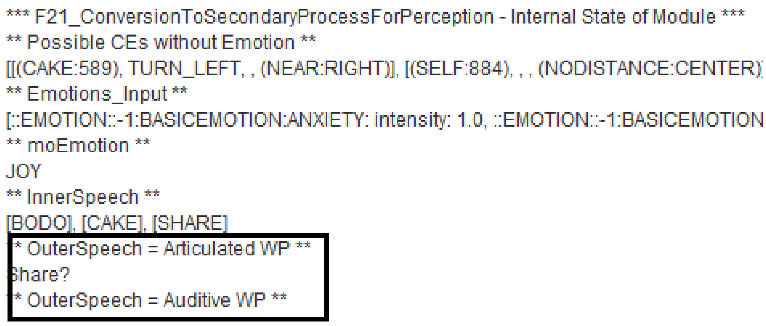

Figure 6.18: Inspectors for Listening

6.3 Situational Context: Ask and Answer

The situational context for asking another agent is established if the following CE categories occur, as it can be seen in fig.6.19. In comparison to the first use case the agent is not hungry and not angry. Regarding the CE: Emotion and the CE: Drive, the Drive is Relax with the drive component libidinous, meaning that he will rather ask in a positive way.

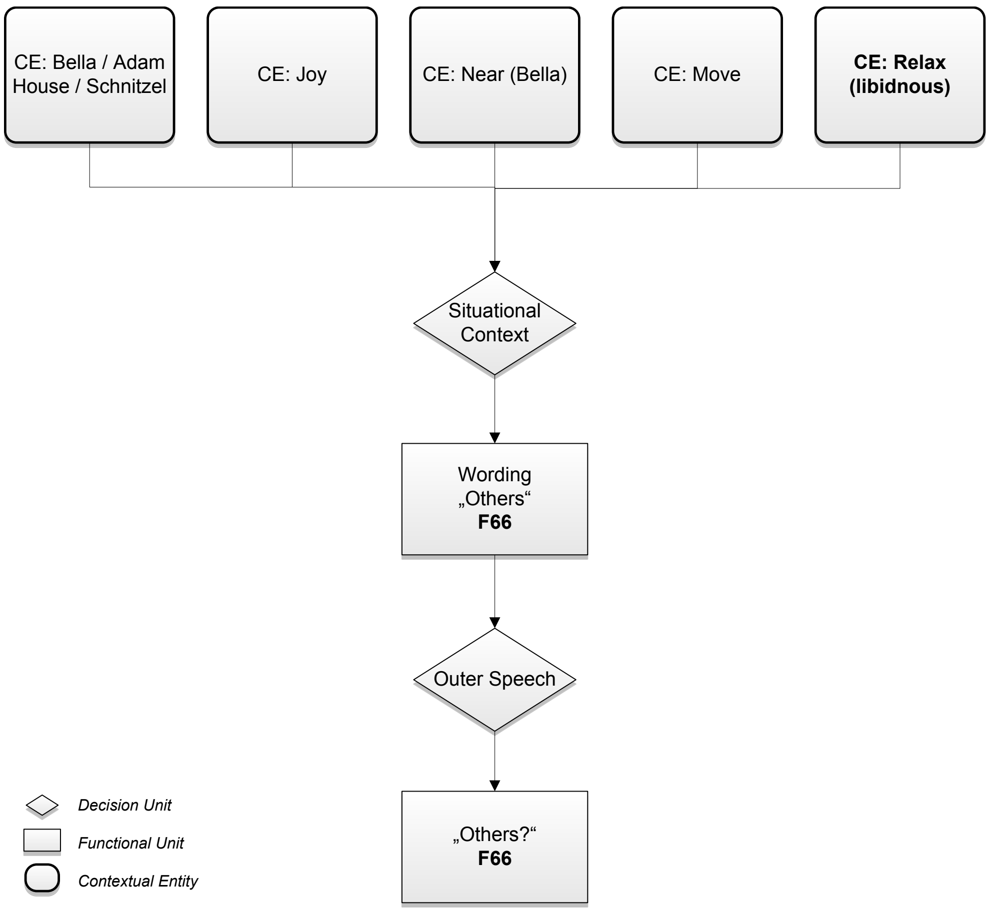

Figure 6.19: Situational Context Ask and Answer

The situational context for answering is established if the following CE categories occur, as it can be seen in fig.6.20. At first the question is received as an auditive WP and is processed accordingly. Only when the situational context for Known is recognized the concept and the wording of known is retrieved. Regarding the CE: Drive, the drive is Relax, which generates a rather positive answer of "Yes."

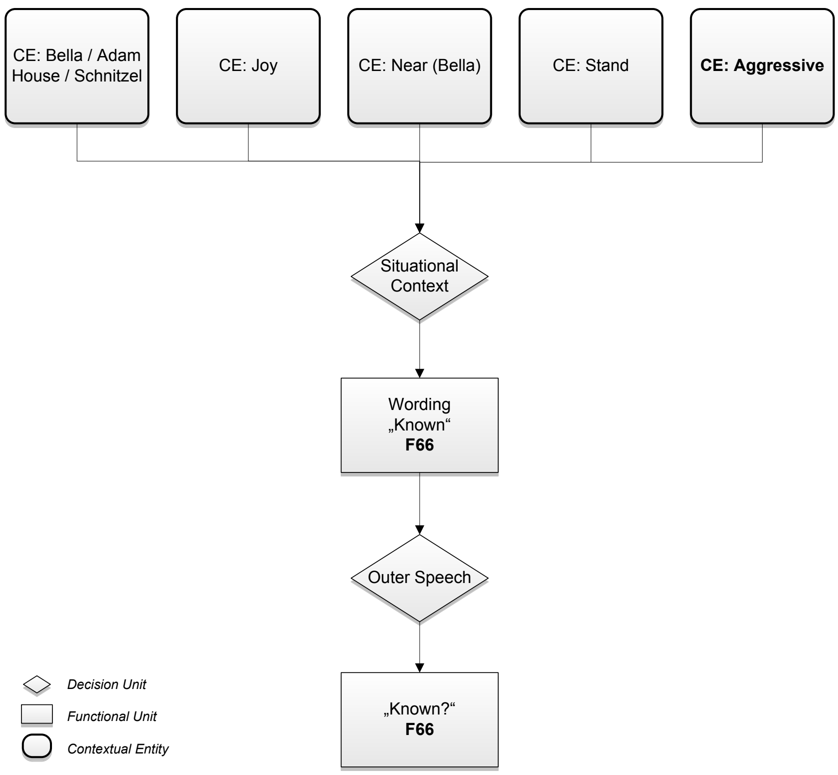

Figure 6.20: Situational Context Answer

All situational contexts that have been represented by now are set in the ontological knowledge base as. For further configuration of the ontology, the structure should be explained at this point. It can be taken as a template and thus can be generalized. To generate a context frame one has to configure the persons and objects in the *object field*. Then the action is set in the *action field*. The field *consequence* describes what happens after a context has been recognized. The field *precondition* sets the drive value, which is essentisal for the wording "Welcome". Last but not least, the anticipation value describes by means of an anticipation value how much it pays off to say something. In the case of saying "Welcome" in this situation, the anticipated pleasure level is 0.7, for instance. It means there is an increase in the pleasure level of 0.7, if the utterance is actually produced. In case there is an utterance in the same situational context that has a value of 0.3, this utterance brings less pleasure than the context with the value of 0.7. The utterance is selected that brings most pleasure in a given situation.

After the situation is recognized by the ontology, inner speech is produced. The CEs in the situation are: Finally, the end result the situation leads to an outer speech statement "What?" as it can be seen in fig.6.21.

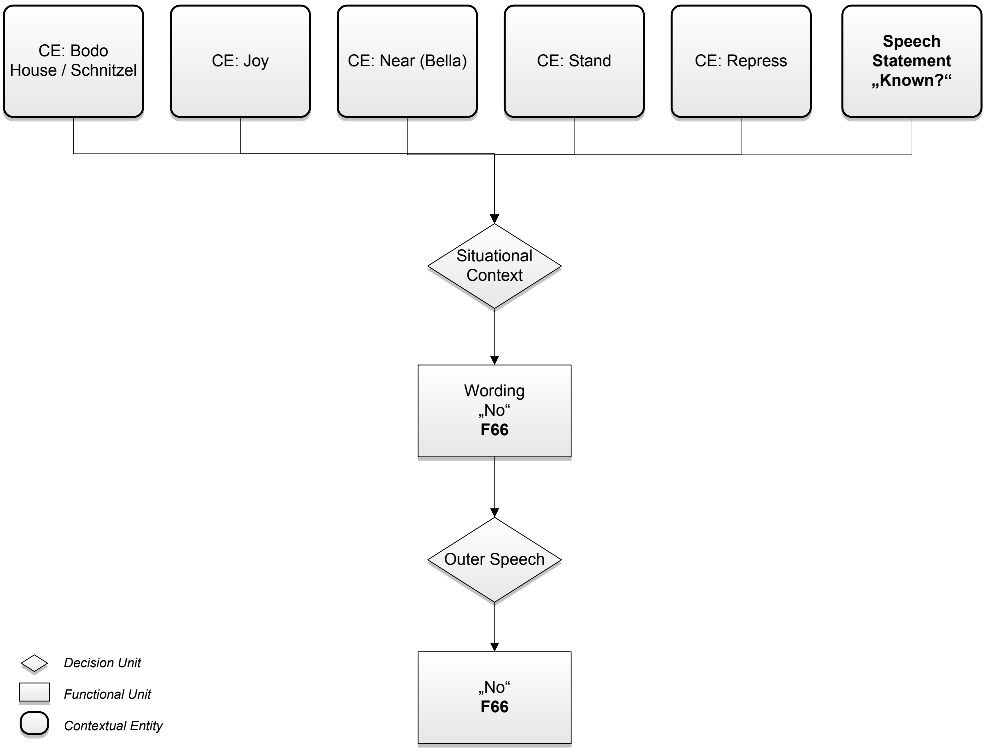

Figure 6.21: Situational Context Ask

6.4 Discussion on the Implementation of Situational Context Recognition

The finding of a situational context situation depends on the match between the memory trace in the ontological knowledge base and the perception information from acoustics and vision. To recapture a match needs to be a full match of all five CEs, meaning every category contains the information of a CE. This has several implications. A full match of all contextual entities is needed for identifying one situational context.

There are several options to improve this bottleneck. The problem is that only after a full match situational context is identified.

One solution to this is that the entities in the memory of the ontology can be equipped with weights and a limit for the recognition of a context is set. Then a situation for example is recognized if 3 out of 5 CEs are given, meaning only sixty percent of the of the situational context has to be fullfilled in order recognize the context.

Another possible solution for the recognition of situational context is that CEs can occur in several different combinations. As seen in table 6.2 seven different contexts for Welcome can exist. The result utterance is always the same "Welcome", although different context situations are described. The reason for it is, that one utterance has slightly different context situations. This approach shows that, though different types of context occur, the outer speech statement is always the same.

Situational Context	CEs	Outer Speech
Context Welcome 1	Bella, Agent, House, Near Bella, Move, Joy, Relax	Welcome
Context Welcome 2	Bella, Agent, House, Move, Joy, Relax	Welcome
Context Welcome 3	Bella, Agent, Near Bella, Move, Joy, Relax	Welcome
Context Welcome 4	Bella, House, Near Bella, Move, Joy, Relax	Welcome
Context Welcome 5	Bella, Agent, House, Near Bella, Joy, Relax	Welcome
Context Welcome 6	Bella, Agent, House, Near Bella, Relax	Welcome
Context Welcome 7	Bella, Agent, House, Near Bella, Joy	Welcome

Table 6.2: Different Ontological Matches for one Context

Usage of Algorithms and their Performance

The used algorithm for the problem of recognition situational context is the greedy algorithm that has been mentioned in chapter *Concept*. The reason for selecting a greedy algorithm is that a lot of elements need to be searched through in order to match them to the categories. The mechanism of finding a situational context is very time intensive. Thus, an optimal algorithm is needed. In the following three different methods for algorithms are presented: the greedy search, constraint-based search and brute force search.

A greedy algorithm is a mechanism that always terminates and makes locally the optimal choice in the hope that this will globally be an optimal solution to the problem. The performance of the greedy algorithm for this problem is O (n), meaning that it has to run through every element once until a match in the search of CEs is found.

In the case of a constraint-based algorithm the strategy is, to set a constraint for the problem of the recognition of context. In the case of the matching problem the solution is to say, when a certain number of CEs are obtained in the CE Working memory then a situation can be recognized. The performance in case of the constraints-based can be better than the greedy algorithm in case the constraint is defined better and finds a set of CEs sooner. An example for constraint-based alorithms is depth search or breath search, which searches until a certain level and then stops.

Another option besides the greedy and the constraint based algorithm, is the socalled brute force method, where the global optimum is selected at each step. The brute force mechanism tries out all eventual candidates and verifies whether the candidate is a match. It is supposed to be an exhaustive search that always terminates and finds a solution. It is well suited for smaller problems. Thus, as long as there are only a few memories the brute-force method can be taken over the greedy method or the constraints-based algorithm.

In conclusion, it can be said that due to the requirement each element needs to be stepped through the greedy algorithm is suited best, because at the moment the number of elements is not exploding. Thus, a greedy solution will find a global optimal solution that way.

6.5 Discussion on Implementation of Inner and Outer Speech

Based on the design and implementation process some points shall be mentioned that can be improved. Simulation and Implementation are treated together as they are entangled with each other.

The programming of the ontology is currently a bottleneck. The disadvantage is that the implementation of the ontology uses one file, which is shared among developers. Due to a the tool Eclipse and the package GIT that is used, the adaption of the ontology is error-prone. It is hardly possibly to work at the same time at the same ontology. In fact the problem can be solved when using another file management system as for example. Another cheaper solution is the integration of owl into java without using Protégé as an external client.

Regarding the design it is practicable to design an ontology that is extensible for later versions. Extensibility of ontologies is achived by designing a general upper level and a domain-specific lower level ontology. The design characteristica is that if using this structure, all parts that belong to the core are part of the general upper ontology. The rest of the classes, which can be extended at any time or deleted at any time can be in the lower level ontology. For example, additional classes, such as objects that are new are anchored in the lower level ontology. However, the attributes needed for registration of situational context, such as place, entity, content are based in the upper ontology.

Regarding version management, the ARS project uses a version management tool called GIT [1] but there is no unique process how to manage the source code. In software engineering, usually revisions are for bigger changes and versions usally deal with a small change. However, the problem in the ARS sourecode is that changes from version to version are large. This is complementary to the approach in software engineering. It would be more usefull to make revision for bigger changes. Apart from that, the usage of alternatives it is possible, indicating that a parallel development takes place. Alternatives are not used at all in the current programming tool GIT.

On the design issue another critical point shall be mentioned. The aspect of time is not integrated in the word statements meaning that agents can not communicate based on temporary aspects. It means that agents can not communicate about time, as an understanding about the concept of time is not integrated in the ARS model [DBM+13]. According to Levelt when forming a thought in a sentence like structure at first the TIME then the EVENT and later on the PLACE occurs. Without the clue of TIME it must be in the order EVENT and PLACE. In the fig.6.22 below it will be shown how the processing takes place by conceptualizing, formulating and articulating. for the TIME, EVENT and PLACE.

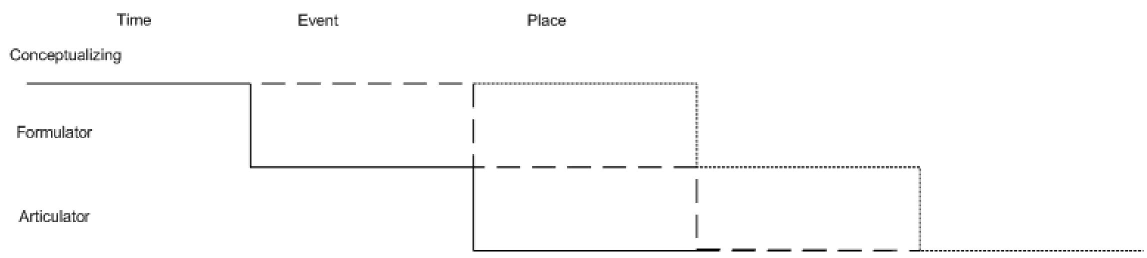

Figure 6.22: Time Event Place Diagram

In the fig.6.23 below it will be shown how the processing takes place by conceptualizing, formulating and articulating. Without time as an additional aspect the order is limited to EVENT and PLACE.

An advantage of the simulation is the fact that is rather simple designed. The simulation can be visualized in a 3D environment as well but the weight in the ARS model is not on a fancy

[1]http://www.git.com.

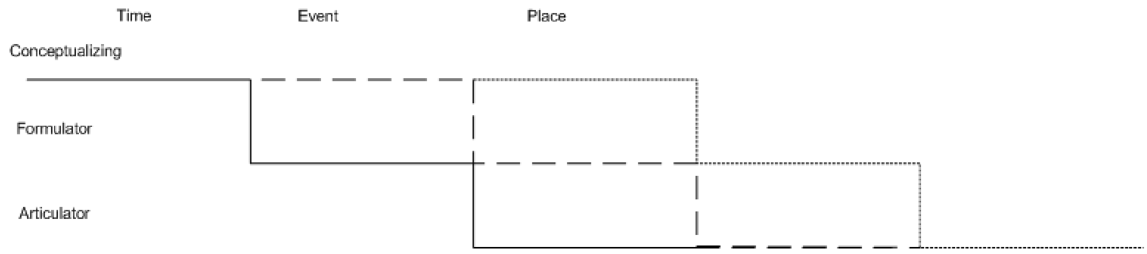

Figure 6.23: Event Place Diagram

graphical design. Instead the aim of the simulation is to show the behavioural output by means of a simple looking multi agent environment. Thus, also in the graphical design of agents the weight is not put on the gestures but more on the actions conducted by the agents as such.

The simulation of situational context is represented by means of speech and overlay action icons that have been introduced in the beginning of this chapter. The advantage of these overlay action icons is that they are easy to visualize within the simulation. The disadvantage is that the customization is not possible at the moment. If a new overlay action icon is required it is constructed manually by using the application paint.

6.6 Discussion on Implementation of Listening and Speech Comprehension

Speech comprehension or listening is the process of looking for the matching visual object association. During the process an auditory word presentation that is received is matched to its object association. The process in the order of generation is at first, the acoustic impression comes in. Then the visual object association is connected to the loud image or word. The link between a visual object association and a wording is exactly between visual and acoustic.

The process listening is initialized when a statement of outer speech is received by an agent. The statement has to be within the radar of the agent receiving the speech statement. This is due to the fact that the agent's ability to listen is not infinite but only in a certain space. Outer speech does not distinguish between near speech and far speech. If speech enters the vision and acoustic radar the it can be perceived and fruther processed. The speech statement thus is an object of speech, that has the property of sound beeing delivered and that has content attached. The content of the speech statement is accessed after the visual object association gets matched with the acoustic loud image.

The work solves the problem of inner and outer speech in a certain situational context. The context is in addition influenced by the drive content of the agent. Speech comprehension is realized by means of a situational recognition of the received word and finding the correct semantic assignment to its visual object. For example, outer speech is received via the word and linked to its visual object. If the matching is achieved semantic content is accessible.

As regards the limitation to reasoning, the following needs to be stated. Inner and Outer Speech may result from reasoning. For instance, a chess player that formulates his strategy verbally. Decision making and Planning are controlled by the drive instance. Speech Production Modules are the articulator, formulator and conceptualizing unit. They are first limited by context and secondly by the drive context. This is the drive the agent actually has in a situation.

7 Conclusions and Outlook

"Wer so tut, als bringe er die Menschen zum Nachdenken, den lieben sie. Wer sie wirklich zum Nachdenken bringt, den hassen sie." Aldous Huxley

The final chapter reflects on the language production model, the selected ontological architecture and makes suggestions for future work, such as the implementation of overt speech in a sentence like form. Furthermore, the advantages of a rather rule-based architecture is dicussed in comparison to a a non rule-based architecture for language production. The introduction *Brave New World* discusses the points of manipulation of people's concept or schemes from the moment they are born and what is the implication for the language following an argument from Lev Vigotsky.

7.1 From Brave New World to Conceptualization

Based on the work at hand, situational context has been applied in a multi-agent scenario, leading to a context-aware speech and thought system called inner and outer speech in this work. Important is however the fact that within a defined frame of situational context, speech and thought can be established. The speech facility enables seeing what an artificial agent thinks and how it communicates in a certain situation.

Speech and thought allow that artificial agents have different concepts about things, due to the conceptualization of objects. For example, an object is perceived as a plant from one agent and as a food source from the other agent, because the second agent is vegetarian and has the concept of plants can be eaten, whereas the first agent has the concept of plant can not be eaten at all. The conceptualization describes the object and its associations that can be perceived by an agent when perceiving an object. Objects that are known, can be retrieved by means of a memory trace by the agent. Also, objects that are unknown can not be retrieved. Thus, the ontology of an artificial agent contains only memory traces that have been experienced and stored in memory.

The novel *Brave New World* [Hux32, p.14] illustrates how imprinted concepts and environmental perception work together. Even more important Aldous Huxley [Hux32, p.30] shows how the world looks like, if concepts of people are built already before they can gain any perception of the world, before they start learning even. In the novel, the idea people are formed and manipulated from their embryonal beginning, which shapes a system of several casts that differ in thought and language [Hux32, p.14]. The idea is to condition people in order to eliminate competitiveness.

On the other hand there is a strict hierarchy in a cast system where alpha people are above beta people for example.

People of their own cast can communicate inclusively due to the fact that they have the same concepts imprinted. A member of the gamma cast can not understand the alpha cast which has a far more reacher set of possibilities of how to devleop creativity, cognitive abilities and so on. *Brave New World* illustrated with exaggeration how conditioning of concepts from a very early way in life can have implications on speech and language.

Transfering the metapher to the ARS model it means the following. The ability to recognize concepts has been implemented for the ARS model by using the socalled TPMs and WPMs. This means ARSins can be understood only exclusively by ARSins. This is due to the fact that all ARSins own the same concepts. If a concept is communicated to someone else but an ARSin one has to make sure that the concept is understood. To ensure this a mapping to be done between different concepts. A valuable help for the understanding of concepts is context itself. Context sets the frame, gives hints and enables people to understand a concept without knowing exactly what is the concept.

Another argument regarding concept formation in children has been postulated by Lev *Vygotsky*. [LEHG12] He studied concept formation mainly in primary schoolars and used two methods. First, the investigation of the child's concept through verbally defining the content. The method looks at the already formed concept. The second method he used was abstraction. The method looks on the other hand on the psychic processes leading to concept building. Vygotsky concentrates on the communicative aspect of speech and claims that the word is a tool of mutual understanding. What happens is that during mutual understanding a sound acquires a certain meaning and becomes a word or concept.

To understand his theory of concepts one has to know how he uses the word sign. Vygotsky sees the word as a generalisation and thus a word becomes a concept. According to Vygotsky: "Words take over the function of concepts and may serve as means of communication long before they reach the level of concepts characteristic of fully developed thought" [L.78]. What Vygotsky means is that the child starts talking to his colleagues about rather the sign or word to comprehend the meaning of the word. During the process of talking to others using the word or sign allows the child to access the meaning.

Following, through social interaction and the community, children and adults learn how to use new concepts appropriately. Vygotsky further claims that new concepts in mathematics work similarily. At first the mathematical concept is developed and in the second place the concept is exchanged with the social community. These two steps lead to the development of a mathematical concept.

Mapped onto the ARS project it means that agents have to talk to each other and use new concepts. By talking to each other they need to clearly exchange the concepts and by talking to each other obtain the meaning of the new concept. Also it would mean consequently that learning of new concepts has to be implemented after an exchange. In case an agent detects a new concept he has to store it in the knowledge base and retrieve it when linking or retrieval is necessary. Linking the concept to another concept happens during verbal exchange. To understand the meaning the concept has to be linked to an existing concept and related to existing other concepts.

A critical side note to the theory of Vygotsky is however, that he does describe in a detailled way how this concept formation works in the brain or neurologically. Meaning, that he describes concept formation in a mereley sociological and empirical way. A description of concept formation based on the cognitive processes that take place could extend his approach.

7.2 Conclusions

The embedding of the language production model within the ARS model has been a rich output for the simulation. By means of inner speech and outer speech it is possible to see the interactions of the agents in communication. Added to that, the usage of an ontological knowledge base enabled the communication of agents and the semantic understanding of words in a certain context.

Thus, this chapter first reviews the consquences of the integrated language production model into the ARS model. Second, the Protégé ontology architecture is critically reviewed and third the selection of the use case is summarized and commented. The last subchapter, selection of the uses case, addresses the selected use cases and suggests further open issues.

The language production model has been adpated and integrated in close collaboration with psychoanalytical experts. Several working iterations and steps were needed to integrate concept formation (see chapter *4.2.1. Concept Formation*) within the model, which is why the conclusion reflects upon concepts. Another challenge was the psychoanalytical embedding of inner and outer speech within the pleasure principle. As defined in the Technical Review in 2013 [DBM+13, p.97] the requirement for outer speech speech is according to the pleasure principle.

A first step towards integration was the evaluation (see chapter *2.4. Evaluation of Language Production*) that has been done. It shows that the model from Levelt can be integrated only with constraints into the ARS model. The constraints are for example architecture related. The model of Levelt is a modular, functional architecture comprising three main layers. The information flow in Levelt's model is from the conceptualizer to the articulator and then from the speech comprehension module to the conceptualizer again.

The ARS model has the functional modules in a certain order. A speech related functional module as F66 (Speech Ability) had to be localized right after the primary process and at the beginning of the secondary process due to the requirement in psychoanalysis that the word presentation is attached to the thing presentation. This is mainly because the primary process is dominated by activated thing presentations, where the word presentations are linked to. The second relevant module for speech production is F53 (Reality Check for Action Planing), which is besides the reality check responsible for checking whether uttering a word brings enough pleasure. In case this module decides, that it is better not to say something, the speech content as a wording is sent back to F46 and makes another iteration. Then the wording should be articulated as a speech statement. The module F30 (Motility control) is responsible for transferring the speech content to the actuators so that the agent is able to speak finally. For the hearing process some adaptions needed to be done in the primary process module F10 (Sensors environment) and F11 (Neurosymbolzation environment) as well. The first requirement is that a statement, that has been articulated is received by another agent. Thus, in the primary process a speech statement is received by environment sensors and then is subsumed with other sensory information. That way, speech input is symbolized, meaning that what we hear as acoustic noise is transformed to a meaningful symbol in the next step.

In summary it can be said, that the challenge was to integrate the speech related modules (F66, F53, F30, F10 and F11) and correctly adapt them to psychoanaltical requirements such as psychic energy access or the pleasure unpleasure principle (see chapter *4.2.1 Modeling of Speech Production*).

7.2.1 Conclusions for the Language Production Model

This chapter continues, where chapter *State of the Art* has left. As the evaluation shows the model of Levelt is situated ideally for the functional oriented ARS model. After implementing parts of the speech production model the question is raised, if the speech production model implemented now is still the same as the original model. The question aims at the content of the modules and is less interested at the architecture as a whole.

For that reason the following modules from the Levelt model are compared to implementation ARSi13:

a) Conceptualizer,

b) Formulator,

c) Articulator,

d) Lexicon.

At first, the conceptualizer module is located within module F66 (Speech Ability) that has been mentioned in the subchapter before. In Levelt's original work [Lev89, p.21] the conceptualizer is situated in the pre-linguistic part of his model, meaning it is not yet a part of the linguistic process, but more a relict of the conceptual memory where it is guided from. In the ARS model however, the conceptualizer module was implemented in the linguistic module for Speech Ability.

The second module is the formulator. The formulating process as Levelt discusses it, takes place in the module F53 (Reality Check for Action Planning). In this module, the pleasure principle is responsible for monitoring, whether the pleasure is high enough that a word should be produced. In the original model from Levelt, the formulator produces internal speech in a twofold process consisting first of grammatic encoding and phonetic encoding. These encoding processes are not required in the speech model integrated in the ARS model, because outer and inner speech is produced by a one to three word statement.

The third module, is the articulator that is located in module F30 (Motility Control). It is responsible for transforming the wording which is attached to the word presentation to an actuator output. If the agents would look more human like and have an actual mouth, it would be important that the motor actuators move according to the word that has been produced. For example, uttering an "O" requires to open the form the mouth in a round way. As this actuator information is not required in the ARS model at the moment, it has not been taken into account for the work. However, in a roboter or avatar with a mouth or face, it has to be taken into account that if an utterance takes place the mouth has to move accordingly.

Apart from that, a lexicon is needed in Levelt's original model it connects the formulator to the speech comprehension unit and stores lemmas of speech. The lexicon in the original model is integrated closely with the speech comprehension unit. In the ARS model the lexicon resembles the Protégé ontology and is not integrated with speech comprehension. The reason for that is that in the ARS model data is separated from functions, which is why speech comprehension functions are implemented in the functional modules and data is treated seperately in a knowledge base.

7.2.2 Conclusions on the Selected Ontology Architecture

As presented in the previous chapters in the *State of the Art*, the ontology architecture has the following advantages.

a) Represent and recognize situational context,

b) Use one ontology for many different situations,

c) Flexibility of the used ontology,

d) The possibility to use greedy mechanisms.

Based on the context entities *Action, Content, Distance, Entity and Drive* the frame is settled for capturing any situational context as possible. The context is represented in the Protégé ontology and can be maintained there manually at the moment. The general ontology contains the CE categories for situational context detection while the domain-specific ontology contains the detailed properties of each CE category. The reason for using this system of context detection is that the environmental context model used in ARS is not very complex at the moment. The simulation for the third use case entails roughly about 20 known objects. Of course the system becomes richer with the memory associated to the objects.

However, another advantage is, that one ontology can detect many different situations, depending on how exact the match is set. If the factor of matching is less accurate there will be more situations that ca be detected instead of when the factory is digital 0 or 1.

The example of using an ontology is, that it can be extended easily in the architecture at hand. Due to the modular construction of the memory traces and the CEs there is an easy way of extending the whole architecture if there is a need to.

Finally, the usage of reasoning mechanism is an advantage. Meaning for obtaining the CEs for the context detection a greedy algorithm is implemented. Therefore, there are different greedy strategies that can be used, such as Depth-First-Search or Breadth-First-Search. What has to be taken care of regarding these algorithms is the order of access, that can be differently.

Furthermore, when comparing constraint based algorithms vs. greedy it is fact that with more complex evaluation methods the runtimes can get larger, if the algorithm compares only one element to maxima or minima. A plausibility check is able to check whether a large value is plausible at all.

7.2.3 Conclusions on the selected Use Case

The use case selected for the work has been based upon the second and the third use case, which is situated best for language requirements, due to the fact that the agents need to communicate there.

Still there are some disadvantages that come with the third use case. For once, the UC has not been detailed enough designed for language implementation and combined psychic conditions. In order to be detailed enough for a technical realization every drive and effect on language has to be laid out before. During the realization of the third use case in the simulation it has been needed to show the drive which is responsible for the outer speech statement. Thus, the need has arised to define a table of drives that are responsible for diverse speech messages. For example, the drive relax-libidinous is responsible for the speech message: "Welcome!", whereas the drive relax-aggressive is responsible for the speech message "Known?", which stands for "Do you know him?".

Use cases on the other hand allow to define requirements and observe them within the use case. By introducing language and speech one requirement was that agents can perceive and receive speech information. Furthermore, they can make a semantic assignment between the speech and its association. Thinking is in general a prestep to outer speech. All these requirements are formulated in the very beginning as hypothesis. In the end when seeing the agent act we observe that kind of behaviour that has been designe in the requirements. The disadvantage of this is that any other requirements that have not been coverd in the start can not be observed, of course.

In order to extend this requirement-based engineering it is possible to implement learning of new concepts. Only when agents are able to learn a concept of their own a new requirement is formed.

7.3 Further Work on the model

The section covers ways how the existing model can be improved. For once, an integration of actions that are based on context could be a worthwile improvement. The generation of actions can be also used to generated alternatives, which will be explained as well. Secondly, eventually at some point the need will arise that agents have to comnunicate more that just on a one word basis. When this arises, one has to think about a whole grammar and a sentence structure. Thirdly, the ontology architecture can be changed depending on the development and the demands on the model. Finally, the appliance of this model is discussed in different fields such as cognitive robotics and automotive industries.

7.3.1 Generation of Actions

Actions are treated as an abstract form of a speech action. Hence, they are generated within the framework as well, whereby a scene encapsulates the situational context. Following, a scene contains only certain context-dependent speech actions and actions. That is due to the fact, that like a speech action, an action is situated within a situational context.

The problem of generating context-depending actions is based on the fact that an action can be a consequence as well. Meaning, if someone hears somehthing the consequence is an action in the form of running and not a verbal action by talking.

As it can be seen in the fig.7.1, a solution to the problem is that actions are modelled as consequences in the Protégé ontology and establishing weights for different consequences.

Action selection takes place in the second iteration of Levelt's model. What actually happens is that the an auditive word presentation is received by the agent. This auditive statement is connected to the visual object presentation.

In fig.7.2 it can be seen that action selection just as word selection is processed in a similar manner. An auditive word presentation is processed and after the word has been recognized in it's context an action can follow. For example, Adams saying "Eat" has as a consequence that Bodo starts

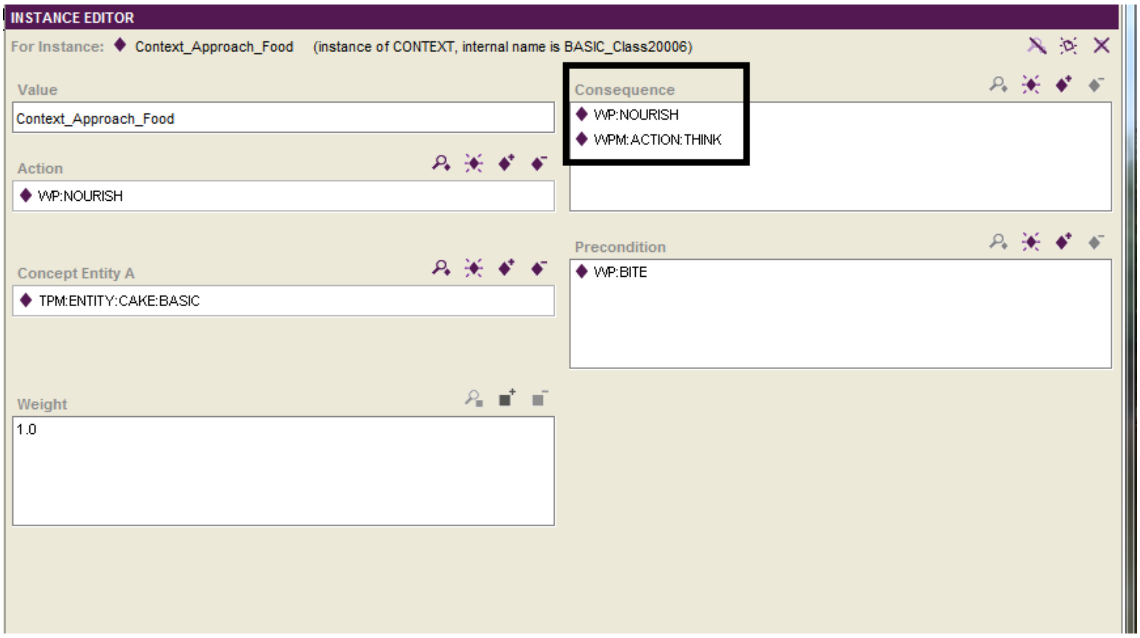

Figure 7.1: Action Selection

eating. In F66, the action Eat is validated against the situational context, meaning it is checked whether an action is appropriate in that context. If an action following a speech statement is appropriate the agent will conduct it. If it is not appropriate he will not conduct it.

Based on the use case two introduced in chapter *Concept* the following actions take place in the following form: Adam takes the food and eats it on his own - the action is eating. Another example is: Adam attacks and bites the other agents and eats the food eventually on his own - the action is attacking and biting.

Apart from this, another problem that should be considered is scheduling of actions. It means, if an action follows another action, a rule for action scheduling has to be designed. This is needed because otherwise the appropriate action selected in module F66 can be interrupted by decision making. The scheduling however, asssures that only adequate actions are executed in time. Thus, the action scheduling list contains a list of adequate actions to be conducted in a certain order.

The perception of other agent's action offers the possibility for perceiving what the other agent does. Thus, at first an understanding of one's own actions has to be given before one can think of the action of someone else. In principle the implementation is simply the action of the other agent marked with the tag action notself.

7.3.2 The role of Outer Speech and Speech Comprehension

Outer Speech in this work is the articulation of a speech statement in a certain context presuming that there was inner speech in the form of thoughts before.

In the following the notion of speech comprehension is discussed. Speech comprehension involves according to Levelt inner speech or outer speech as an input. Meaning that thoughts and articulated speech is needed for a speech comprehension system. Apart from the fact, that Levelt has

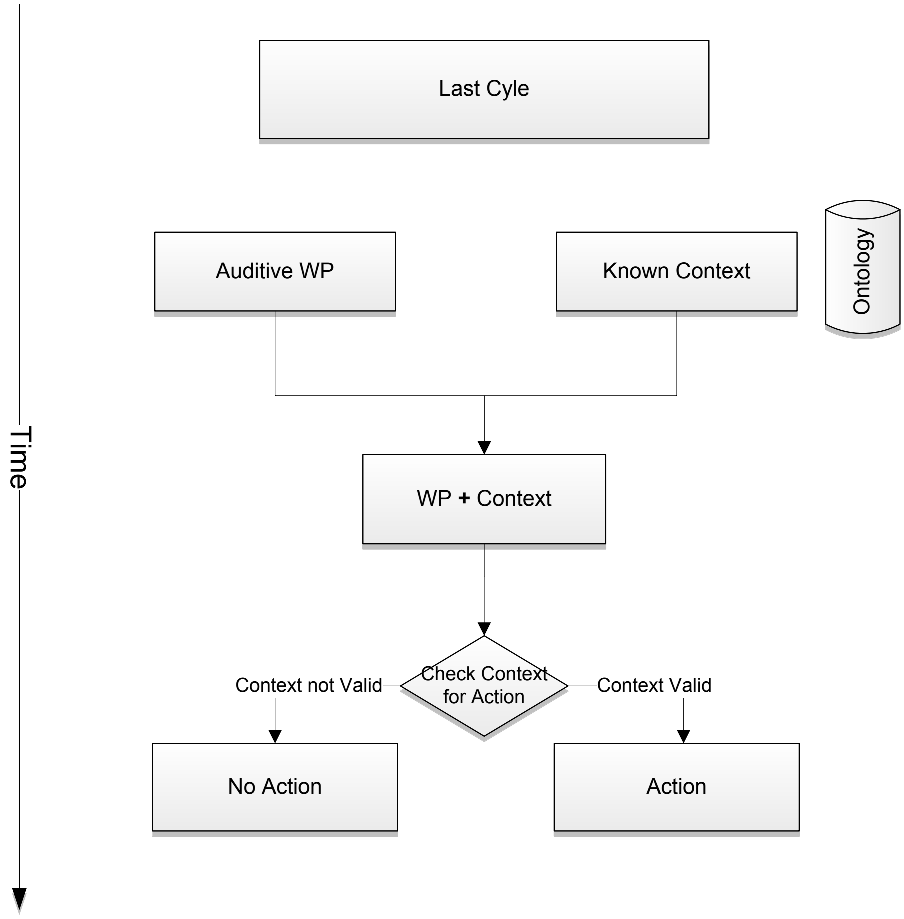

Figure 7.2: Action Selection projected onto Levelt's model

left out the speech comprehension, he postulated that speech comprehension needs access to the lexicon containing the lexicon lemmas and forms as it can be seen in fig.7.3 below.

The term *speech comprehension* is defined as the actual understanding of words that have been generated. Understanding of a word thus is the word selection in dependence of the situational context and the drive. To understand a word the prequsites are a semantic database of outer speech statements resembling to Levelt's lexicon, a mechanism to retrieve the speech statements its concepts and wordings, and a mechanism to make associations to retrieved words in a certain context.

Speech comprehension in this work is addressed merely from the secondary process during the semantic assignment of the wording to the situational context. However, the process of speech comprehension starts in the primary process by means of integration of the acoustic speech data from the perception. The meaning of an object presentation is set when the word presentation is attached to the object presentation. For example the loud-speech object presentation <mamaaah> becomes the meaning Mama by means of the word presentation <Mama>.

Following, speech comprehension in the ARS model depends on the access to the knowledge base of the words and the speech production module where the loud image is attached to the thing

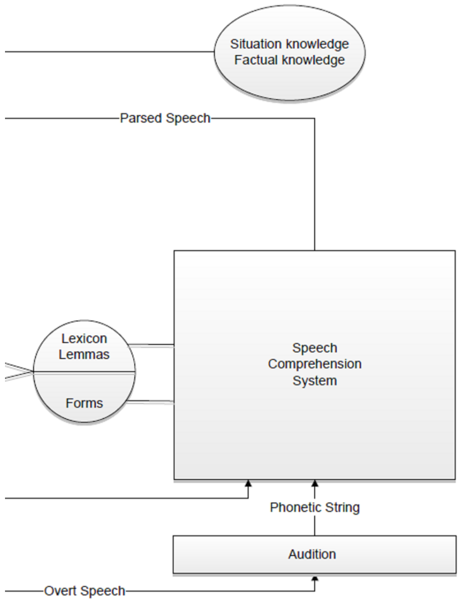

Figure 7.3: Focus on the Speech Comprehension Module in [Lev89]

presentation. The semantic meaning can be changed, if the assignemt in the speech produce module is manipulated. For example, if the acoustic loud image is processed before the visual object association and the context, which happens when listening to someone. In this case ambiguitiy is produced if the listener can not distinguish the bank to sit from the bank to fetch money.

7.3.3 Future Fields of Appliance

Potential fields for applications using situational context is speech recognition, cognitive robotics and automotive industries.

Cognitive Robotics

For context modeling in ubiquitous environments these environments are modeled using an ontology and rules for matching the contextual concept. Already several ubiquitous architectures exist, that employ context modeling in robotics. For instance, context modeling using ontologies and rules for reasoning is used in intelligent robot environments [GS05].

Apart fromt that, roboters use a context based vision for localization tasks, whereby the robot collects contextual information about an image in a short amount of time [CS05]. Siagian and his colleagues have developed a mobile robotics application that is able to classify the major points of a scene outdoors. In the context based approach the scene is taken in as a whole and a low-level feature is extracted.

An idea of improvement in the field cognitive robotics, especially in pattern recognition and computer vision is that after recognising a scene a warning sound or speech is given as an output, depending if it is a factory or a service robot. For service robots used by humans, ideally the sound of human language is much more naturally than just using a simple sound. Thus, in situations where a user needs to be warned to do somehthing a voice interrupts. Depending on the environment or situation, different voices or speech sounds can be used. For example, in a home for elderly people, a service robot may be used to detect the pattern of a person moving. When someone walks unstable the voice can interact by talking directly at the person and addressing the unstable walk or even suggesting to make a break.

Drive Assistance and Situation Recognition

Another application for recognition of context comes from the automotive sector and the so-called field of driving assistance systems. The recognition of context here is quite important to recognize potential, dangerous situations. Models that use environmental models already exist up-to-date although at the moment environmental models are less complex than reality. As a consequence, driving on the highway is easier than driving in the city due to the fact that more information is needed to cover those scenes.

There are several aspect regarding driving assistance systems that can be optimized. For example, driving autonomously, without the full attention of the driver is currently a topic of investigation. There are several demands on such driving assistance systems ranging from performance, storage to usability needs. Most of the challenges are hardware and software related e.g. hardware devices are getting smaller and thus software has to be smaller of capacity but on the other hand even more functions need to be implemented.

Computing performance for example, is a bottleneck at the moment, as the calculating power for situation recognition is quite high. Due to the fact, there are several on board computers at the moment for evaluating the situation and calculating the driving path. Another disadvantages is the capacity demanding storage of road maps and other maps that are needed for navigation tasks. Different ways of storing map information and data are taken into account in current research. Usability is also of concern regarding safety reasons mainly. For example, the actuator output e.g. visual, acoustic, haptic is very often not synchronized. A reason for the lack of synchronisation is that different signals and functions support the driver but the functions are quite standing alone solutions. For instance, the park assistance and urban ACC devices could be integrated in a fluent manner.

Another facility of usability is the way sounds are presented to the driver. Different sounds are used for a certain alarm or just for information sources. The concept of auditory displays gets more importance in rather visual oriented environments, for example during the guiding of the car. Sounds and tunes are interpreted differenly depending on their affect and meaning. For example, a high pitched noise means that something is rather important. A sound that gradually changes its pitch tone indicates that importance is rising or declining. A low pitched sound may be semantically attached to less important events.

In general, it can be said that automotive industries can profit from using a context depending situation recognition as it has been described and implemented in this work at hand.

Literature

[Aam94] Aamodt, A.: Explanation-driven case-based reasoning. In: *in S. Wess, K. Althoff, M. Richter (eds.): Proc. of Topics in Case-based reasoning* (1994), S. 274–288 28

[AC75] A.M. Collins, E.F. L.: A Spreading Activation Theory of Semantic Processing. In: *Psychological Review* (1975) 23

[AL88] Anderson, J.R. ; Lebiere, C.: *The atomic Components of Thought*. Lawrence Erlbaum, 1988. – 490 S. 29

[al.99] al., M. d'Inverno e.: A formal specification of dMARS. In: *Intelligent Agents* 1365 (1999) 29

[A.R89] A.R., Damasio: The brain binds entities and events by multiregional activation from convergence zones. In: *Neural Computation* 1 (1989), Nr. 3, S. 123–132 25

[AWJ95] A.Th.Schreiber ; Wielinga, B.J. ; Jansweijer, W.H.: The KACTUS View on the O Word. In: *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing* (1995) 20, 21

[AWMB13] A. Wendt, F. G. S. Schaat S. S. Schaat ; Muchitsch, C. ; Bruckner, D.: Usage of spreading activation for content retrieval in an autonomous agent. In: *IECON 2013-39th Annual Conference of the IEEE* (2013), S. 6672–6677 72, 92

[BA91] Bechtel, W. ; Abrahamsen, A.: *Connectionism and the Mind: Parallel Processing, Dynamics and Evolution in Networks*. second edition. Blackwell Publishers, 1991 29

[Bal11] Ball, J.T.: Explorations in ACT-R Based Cognitive Modeling?Chunks, Inheritance, Production Matching and Memory in Language Analysis. In: *Advances in Cognitive Systems: Papers from the 2011 AAAI Fall Symposium* (2011) 29

[BHCC09] Benta, K.-I. ; Hoszu, A. ; Cariu, L. V. ; Cret, O.: Agent based smart ?house platform with affective control. In: *Euro American Conference on Telematics and Information Systems* (2009), S. 1–7 35

[BR07] Baumgartner, N. ; Retschitzegger, W.: Of situations and their neighbors mevolution and similarity in ontology-based approaches to situation awareness. In: *Proc. of the 6th International and Interdisciplinary Conference on Modeling and Using Context* (2007) 13, 15, 28

[Bra87] BRATMAN, M.: *Intention, plans, and practical reason.* Harvard University Press, 1987 28

[BVH11] BRUCKNER, D. ; VINZCE, M. ; HINTERLEITNER, I.: Towards reorientation with a humanoid robot. In: *Proceedings of the 1st ISoLA workshop on Software Aspects of Robotic Systems.* (2011) 3

[CF04] CLÚA, O. ; FELDGEN, M.: Differences among Experts, Novices and Trainees Writing a Report. In: *34th ASEE/IEEE Frontiers in Education Conference* (2004), S. 20–23 22

[CFJ05] CHEN, H. ; FININ, T. ; JOSHI, A.: The SOUPA Ontology for Pervasive Computing. In: *Ontologies for Agents: Theory and Experiences* (2005), S. 233 – 258 16, 17

[CGDB97] CHANG, F. ; GRIFFIN, Z. M. ; DELL, G.S. ; BOCK, K.: Modeling structural priming as implicit learning. In: *Computational Psycholinguistics* (1997) 32

[Cha95] CHANDLER, D.: The Sapir-Whorf Hypothesis. (1995). `http://www.aber.ac.uk/media/Documents/short/whorf.html` 69

[CPC+04] CHEN, H. ; PERICH, F. ; CHAKRABORTY, D. ; FININ, T. ; JOSHI, A.: Intelligent agents meet semantic web in a smart meeting room. In: *Proc. 3rd Int. Conf. on Autonomous Agents and Multiagent Systems* (2004), S. 854?861 35

[CQ69] COLLINS, A. M. ; QUILLIAN, M. R.: Retrieval time from semantic memory. In: *Journal of Verbal Learning and Verbal Behavior* 8 (1969), S. 240–247 22, 23

[CS05] C. SIAGIAN, L. I.: Gist: A Mobile Robotics Application of Context-Based Vision in Outdoor Environment. In: *In: Proc. IEEE-CVPR Workshop on Attention and Performance in Computer Vision* (2005), S. 1–7 124

[DBM+13] DIETRICH, D. ; BRUCKNER, D. ; MUCHITSCH, C. ; WENDT, A. ; SCHAAT, S.: *Naturwissenschaftliches psychoanalytisches Modell der Psyche.* `http://publik.tuwien.ac.at/files/PubDat_217802.pdf`. Version: April 2013 4, 9, 36, 38, 51, 66, 67, 85, 92, 104, 113, 118

[Del86] DELL, G.S.: A spreading activation theory of retrieval in sentence production. In: *Psychological Review* (1986), S. 283–321 9, 31, 73

[Deu11] DEUTSCH, T.: *Human Bionically Inspired Autonomous Agents*, Vienna University of Technology, Diss., 2011 1, 77

[Dey00] DEY, A.K.: *Providing Architectural Support for Building Context-Aware Applications*, Georgia Institute of Technology, Diss., 2000 14

[Dey01] DEY, A.K.: Understanding and Using Context. In: *Personal and Ubiquitous Computing* (2001). `http://repository.cmu.edu/hcii/34` 14

[DS00] DIETRICH, D. ; SAUTER, Thilo: 3rd IEEE International Workshop on Evolution Potentials for Fieldbus Systems. In: *In Factory Communication Systems* 1 (2000), S. 145 1

[DSM+97] DELL, G. S. ; SCHWARTZ, M. F. ; MARTIN, N. ; SAFFRAN, E. M. ; GAGNON: Lexical access in aphasic and nonaphasic speakers. Psycholog-evaluation should be merely the first step in improving the model. In: *Computational theories of cognitive behavior allow a precision offical Review* 104 (1997), S. 801–838 30, 31

[Elm88] ELMAN, J. L.: Finding Structure in Time. In: *CRL Tech Report 8801* (1988) 25, 26

[E.T92] E.TULVING, D.L. S.: Priming and memory systems. In: *Neuroscience Year: Supplement 2 to the Encyclopedia of Neuroscience* 1 (1992), S. 130–133 22

[FB07] F. BELLIFEMINE, D. G. G. Caire C. G. Caire: Developing Multi-agent Systems with JADE. In: *Wiley Series in Agent Technology* (2007) 27

[F.D12] F.D., Lars-Åke Álvaro; B.E.Clara; F. Álvaro; B.E.Clara; F.: Erlang as an implementation platform for BDI languages. In: *Association for Computing Machinery* (2012) 29

[FM91] FARAH, M. J. ; MCCLELLAND, J. L.: A computational model of semantic memory impairment: Modality specificity and emergent category specificity. In: *Journal of Experimental Psychology* 120 (1991), Nr. 4, S. 339–357 25, 27

[Fre91] FREUD, Sigmund: *Zur Auffassung der Aphasien: Eine kritische Studie*. Fischer Taschenbuch Verlag, 1891 9, 69, 83

[Fre15a] FREUD, Sigmund: *Letter from Sigmund Freud to C. G. Jung, November 21, 1909*. 1915 (The Freud/Jung Letters: The Correspondence Between Sigmund Freud and C. G. Jung). – 265–267 S. 69

[Fre15b] FREUD, Sigmund: *On the History of the Psycho-Analytic Movement, Papers on Metapsychology and Other Works*. Bd. XIV (1914-1916): *The Unconscious*. 1915. – 159–215 S. 2, 10

[Fre25] FREUD, Sigmund: The Ego and the Id. In: *The Standard Edition of the Complete Psychological Works of Sigmund Freud* Volume XIX (1923): The Ego and the Id and Other Works (1925), S. 1–66 1, 9

[GB08] GALLEGUILLOS, C. ; BELONGIE, S.: Context Based Object Categorization. In: *Technical Report* (2008), S. CS2008–0928 13

[Gib79] GIBSON, J.J.: *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979 3

[GP10] GOBET, F. ; P.LANE: The CHREST Architecture of Cognition. The Role of Perception in General Intelligence. In: *Advances in Intelligent Systems Research* (2010) 29

[Gru93] GRUBER, T.R.: *A Translation Approach to Portable Ontology Specification*. 1993 6

[GS05] GO, Y. C. ; SOHN, J.: Context modeling for intelligent robot services using rule and ontology. In: *in Proc. 7th ICACT* 2 (2005), S. 813–816 124

[GSDG93] G. S. DELL, C. J. ; GOVINDJEE, A.: Structure and content in language production: A theory of frame constraints in phonological speech errors. In: *Cognitive Science* 17 (1993), S. 149–195 30

[Her07] Hernandez, N.: Modeling Context Through Domain Ontologies. In: *Information Retrieval* 10 (2007), S. 143–172 1, 7

[Hin81] Hinton, G.E.: Implementing Semantic Networks in Parallel Hardware. In: *Parallel Models of Associative Memory* 8 (1981), S. 161–187 24

[Hin09] Hinterleitner, I.: Connectionist modelling of language morphology in Williams Syndrome. In: *Language and Cognition 2009* (2009) 73

[HOM04] H., Knublauch ; O., Dameron ; MA., Musen: Weaving the biomedical semantic web with the Protégé OWL plugin. In: *International Workshop on Formal Biomedical Knowledge Representation* (2004) 41, 77

[HSB+11] Hoffart, J. ; Suchanek, F. M. ; Berberich, K. ; Lewis-Kelham, E. ; Melo, G. de ; Weikum, G.: YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages. In: *in Proc. of WWW 2011, Hyderabad, India* (2011) 27

[HTA82] H., Chen ; T., Finin ; A., Joshi: Psychoanalytic Listening?With Special Reference to the Views of Langs. In: *Contemp. Psychoanal.* 18 (1982), S. 613–634 70

[HTA03] H., Chen ; T., Finin ; A., Joshi: An ontology for context-aware pervasive computing environments. In: *The Knowledge Engineering Review* 18 (2003), S. 197–207 19

[Hux32] Huxley, A.: *Brave New World*. Chatto and Windus, 1932 116

[K.98] K., Doblhammer: *Das Sprechen der Sprache*, University of Vienna, Diss., 1998. – 105 S. 69

[KM91] K., Plunkett ; Marchman, V.: U-shaped learning and frequency effects in a multi-layered perception: Implications for child language acquisition. In: *Cognition 38.1* (1991), S. 43–102 73

[L.78] L., Vygotsky: Uznadze. (1978) 117

[Lai08] Laird, J.: Extending the SOAR Cognitive Architecture. In: *Frontiers in Artificial Intelligence and Applications* (2008) 28

[Lan10] Lang, R.: *A Decision Unit for Autonomous Agents Based on the Theory of Psychoanalysis*, Vienna University of Technology, Diss., 2010 1

[LEHG12] L., Vygotsky ; E. Hanfmann G., Vakar: *Thought and language*. 2012 83, 117

[Lev89] Levelt, W.: *From Intention to speaking*. 1989 7, 32, 33, 63, 65, 66, 119, 124

[Lis09] List, E.: *psychoanalyse*. facultas wien, 2009 51, 52

[MLH06] M. Laclavik, M. B. Z. Balogh B. Z. Balogh ; Hluchy, L.: AgentOWL: Semantic Knowledge Model and Agent Architecture, Computing and Informatics. In: *Computing and Informatics* 25 (2006), Nr. 5, S. 419–439 27

[Muc13] Muchitsch, C.: *Human-like perception for psychoanalytically inspired reasoning units : perceptual information processing concepts realized in the ARSi12 framework implementation*, Vienna University of Technology, Diss., 2013 1

[OA97] Oetztuerk, P. ; Aamodt, A.: Towards a model of context for case-based diagnostic problem solving. In: *Proceedings of the interdisciplinary conference on modeling and using context* (1997), S. 198–208 20, 21

[PBA99] P. Busetta, A. H. R. Ronnquist R. R. Ronnquist ; A.Lucas: JACK Intelligent Agents - Components for Intelligent Agents in Java. In: *AgentLink News* (1999) 29

[Rao96] Rao, A.S.: AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In: *MAAMAW* (1996) 29

[RHW86] Rumelhart, D.E. ; Hinton, G.E. ; Williams, R.J.: Learning internal representations by error propagation. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* 1 (1986), S. 318–362 24

[RKIK11] Reinisch, C. ; Kofler, M.J. ; Iglesias, F. ; Kastner, W.: ThinkHome: Energy Efficiency in Future Smart Homes. In: *EURASIP Journal on Embedded Systems* (2011) 2, 35

[RM04] Rogers, T.T. ; McClelland, J.L.: *Semantic Cognition: A Parallel Distributed Processing Approach*. Cambridge, MA: MIT Press, 2004 25

[Roe97] Roelofs, A.: The WEAVER Model of Word-form Encoding in Speech Production. In: *Cognition* 64 (1997), S. 249–284 30, 74

[Rog08] Rogers, T.T.: Computational models of semantic memory. In: *For inclusion in: The Cambridge Handbook of Computational Cognitive Modeling* (2008) 22, 23

[RP08] Retkowitz, D. ; Pienkos, M.: Ontology-based configuration of adaptive smart homes. In: *in Proc. 7th Workshop on Reflective and adaptive middleware* (2008), S. 11?16 35

[Sch00] Schmidt, A.: Implicit Human Computer Interaction Through Context. In: *Personal Technologies* 4 (2) (2000), S. 189–223 13

[SKW07] Suchanek, F. M. ; Kasneci, G. ; Weikum, G.: YAGO: A Core of Semantic Knowledge. In: *Proceedings of the 16th international conference on World Wide Web* (2007), S. 697–706 28

[SLS04] S. Luke, L. P. C. Cioffi-Revilla ; Sullivan, K.: MASON: A New Multi-Agent Simulation Toolkit. In: *Expert Systems With Applications* (2004) 76

[SS05] S. Skaff, H. C. A.A. Rizzi R. A.A. Rizzi: A context-based state estimation technique for hybrid systems. In: *Robotics and Automation* (2005) 20

[ST94] Schilit, B. ; Theimer, M.: M. Disseminating Active Map Information to Mobile Hosts. In: *Personal Technologies* 8(5) (1994), S. 22–32 13

[ST02] Solms, Mark ; Turnbull, Oliver: *The Brain and the Inner World: An Introduction to the Neuroscience of Subjective Experience*. Karnac/Other Press, Cathy Miller Foreign Rights Agency, London, England, 2002 22

[THD04] T., Gu ; H.K., Pung ; D.Q., Zhang: A middleware for building context-aware mobile services. In: *Proceedings of IEEE Vehicular Technology Conference (VTC)* (2004) 17, 18

[Tul85] Tulving, E.: Memory and Consciousness. In: *Canadian Psychology/Psychologie Canadienne* 26 (1985), S. 1–12 54

[Var02] Varchavskaia, P.: Behavior-Based Early Language Development on a Humanoid Robot. In: *Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems* (2002), S. 149 – 158 29

[Vel08] Velik, R.: *A Bionic Model for Human-like Machine Perception*, Vienna University of Technology, Diss., 2008 1, 72

[VSZ06] V. Schickel-Zuber, B. F.: Inferring user's preferences using ontologies. In: *Proceedings of the 21st national conference on Artificial intelligence* (2006), S. 1413–1418 54

[WGDP04] Wang, X. ; Gu, T. ; D.Q.Zhang ; Pung, H.K.: Ontology Based Context Modeling and Reasoning using OWL. In: *Workshop on Context Modeling and Reasoning at 2nd IEEE International Conference on Pervasive Computing and Communication* (2004) 14, 15

[Wid97] Widmer, G.: Tracking context changes through meta-learning. In: *Machine Learning* (1997), S. 259–286 20

[Wit53] Wittgenstein, L.: *Philosophical investigations*. Oxford: Blackwell, 1953 23

[YL06] Yau, S. S. ; Liu, J.: Hierarchical Situation Modeling and Reasoning for Pervasive Computing. In: *Proceedings of 3rd Workshop on Software Technologies for Future Embedded & Ubiquitous Systems* (2006) 15, 16

[Zei10] Zeilinger, H.: *Bionically Inspired Information Representation for Embodied Software Agents*, Vienna University of Technology, Diss., 2010 1, 51, 77

Curriculum Vitae

Curriculum Vitae

Name: Isabella Hinterleitner
Date of Birth: 10.11.1980
Nationality: Austrian

Education	**Time Period**	**Description**
Technical University Vienna	2010 - 2014	PhD Study in Electrical Engineering
University of Birmingham	2008 - 2009	PhD Track (1st Year) Psycholingustics
University of Vienna	2007 - 2010	Master of Cognitive Sciences
Technical University Vienna	01/06 - 07/06	Grant from the Medical University for Dipl. Thesis
Charles University Prague	2005 - 2006	Erasmus study in Prague
Technical University Vienna	2005 - 2006	Master of Medical Computer Sciences
Technical University Vienna	2006 - 2007	Master of Informatics Management
Technical University Vienna	2003 - 2005	Bachelor of Medical Computer Sciences

Work	**Time Period**	**Description**
Robert Bosch GmbH	03/14 -	Researcher Autonomous Driving
Robert Bosch AG	06/12 - 02/14	Role: Software System Designer
Inst. of Computer Technology Vienna	11/09 - 05/11	Project Assistant
AIT Austrian Inst. of Technology	03/11 - 07/11	C++ Programmer
Academy of Sciences Vienna	05/09 - 07/10	VB Programmer
Center for Biomedical Engineering and Physics	01/08 - 10/08	Research (FWF) position
ABB Vienna	11/06 - 12/07	SAP FI/CO Consultant

Language Skills	
German	native language
English	advanced FCE level (written, oral and understanding)
Czech	advanced university entry level
French	basic or indermediate level (written and understanding)
Italian	basic level (written and understanding)