

# **COGNITIVE LANGUAGE ENGINEERING**

## **Towards Robust Human-Computer Interaction**

THÈSE N° 2630 (2002)

PRÉSENTÉE À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

DANS LE DOMAINE DE L'INFORMATIQUE

PAR

**Vincenzo PALLOTTA**

Dottore in Scienze dell'Informazione  
et de nationalité italienne

acceptée sur proposition du jury:

Prof. G. Coray, directeur de thèse  
Dr A. Ballim, rapporteur  
Prof. R. Delmonte, rapporteur  
Dr M. Rajman, rapporteur  
Prof. Y. Wilks, rapporteur

Lausanne, EPFL  
2002

# Abstract

Intelligent information processing seems to be one of the most challenging task among those involved in *human-computer interaction*. A central issue is how to model the various types of interaction among artificial and natural entities at different levels of abstraction. On the one hand, models of interaction are required to better understand the communication phenomena. On the other, suitable languages and paradigms should provide powerful frameworks for developing computer-based applications.

In this dissertation I focus on different aspects of the second problem, trying to develop a methodology for the design of *interactive natural language applications* (e.g. from question-answering to mixed-initiative dialogue). One of the main aspects I am concerned with in this work is the problem of their *robustness*. Several methods have been proposed for achieving robustness in natural language understanding, but these methods are sometimes hard to scale up or re-use in different applications. Moreover, they often concentrate on a single linguistic level of the processing rather than offering a global solution. I will set up a *Language Engineering* environment whose goal is to combine *software engineering* and *cognitive aspects* (e.g. aspects related to representation of a mental model of the speaker).

Given its complexity, it is apparent that the problem can be solved only partially. I want to stress here that the main contribution of my work is a holistic perspective on the problem of natural language understanding. Rather than focusing on a particular aspect of natural language processing, I tried to benefit from the big amount of work that has been already done in Computational Linguistics and Computer Science merging different ideas and techniques.

In the first part of the dissertation I explore the universe of Language Engineering in order to clarify how my contribution can be situated. After a survey on the state of the art on robust methods in analysis of natural language data, I focus on the role that Computational Logic plays in relating the syntactic and semantic analysis of natural language to its practical understanding within specific applications. Robustness is considered from two complementary perspectives, borrowing the terminology from modern software engineering: robustness “in the small” and robustness “in the large”.

The first perspective is discussed while presenting an application for the Interaction through Speech with Information Systems, where robust semantic parsing is used to extract queries from spoken natural language utterances. The second perspective is exemplified by the re-engineering of an existing text analysis system using a new Language Engineering methodology: Agent-Oriented Language Engineering.

In the second part of the thesis I discuss how cognitive aspects can be integrated into a Language Engineering environment leading to the notion of Cognitive Language Engineering. I tackle the difficult problem of *robust dialogue management* from both a cognitive and computational perspective. I propose two frameworks for the semantic representation and assimilation of information into the *dialogue information state*. The first framework allows us to represent and reason about the *dynamic aspects* of objects and events. The second framework is centered on the notion of *mental space* and it is used to build representations of the cognitive processing of information during communication.

# Riassunto

Il trattamento intelligente dell'informazione sembra essere uno degli aspetti più stimolanti tra quelli legati all'interazione tra l'uomo e la macchina. Una questione di centrale importanza è la modellazione di vari tipi di interazione tra entità naturali e artificiali a differenti livelli di astrazione. Da una parte, sono richiesti dei modelli di interazione per meglio comprendere il fenomeno comunicativo. Dall'altra, dei linguaggi e dei paradigmi appropriati dovrebbero assicurare degli ambienti di sviluppo di applicazioni computazionali particolarmente raffinati e potenti.

In questa dissertazione mi sono concentrato sui vari aspetti legati al secondo problema, cercando di sviluppare una linea metodologica per la progettazione di *applicazioni interattive basate sull'uso del linguaggio naturale*, come ad esempio, i sistemi di *domanda-risposta* e i sistemi di *dialogo a iniziativa mista*. Uno degli aspetti fondamentali su cui ho prestato particolare attenzione è il problema della *robustezza*. Pur essendo stati proposti svariati metodi per raggiungere tale obiettivo, questi metodi sono spesso difficili da portare su larga scala o da riutilizzare in altre applicazioni. Inoltre, tali metodi sono adatti a trattare un singolo livello linguistico e non offrono una soluzione globale. Si propone quindi un ambiente per il *Language Engineering* il cui obiettivo è quello di conciliare sia aspetti di *software engineering* che aspetti di tipo *cognitivo*, come per esempio la rappresentazione del modello mentale del parlante nel contesto di un dialogo.

Data l'evidente complessità, risulta chiaro che il problema può essere risolto soltanto in misura parziale. Mi sembra opportuno evidenziare a questo punto, che il contributo maggiore del mio lavoro è dato da una prospettiva *olistica* nei confronti del problema della *comprensione del linguaggio naturale*. Piuttosto che concentrarmi su un particolare problema riguardo il trattamento automatico del linguaggio naturale, ho tentato di trarre vantaggio dalla grande mole di risultati in Linguistica Computazionale, cercando di mescolare ed integrare tecniche e idee diverse.

Nella prima parte della tesi, si esplora l'universo del Language Engineering con il fine di meglio situare il mio contributo. Dopo una rassegna sullo stato dell'arte circa i metodi robusti per l'analisi del linguaggio naturale, mi concentro sul ruolo importante che la Logica Computazionale riveste nell'integrare l'analisi sintattica e semantica per raggiungere l'obiettivo della comprensione del linguaggio naturale in applicazioni specifiche. Il concetto di robustezza viene quindi considerato da due prospettive complementari. Prendendo in prestito la terminologia della moderna ingegneria del software, si considera la robustezza "in the small" e la robustezza "in the large".

La prima prospettiva è analizzata mediante la presentazione di una particolare applicazione per l'*accesso vocale a un sistema informativo*, in cui la tecnica di *parsing semantico robusto* è stata adottata per l'estrazione di richieste (verso basi di dati) da enunciati in linguaggio naturale parlato. La seconda prospettiva è esemplificata attraverso la reingegnerizzazione di un'applicazione esistente per l'analisi del testo, facendo uso di una nuova metodologia per il Language Engineering: *Agent-Oriented Language Engineering*.

Nella seconda parte della tesi, si discute dell'integrazione degli aspetti cognitivi nel Language Engineering, fino ad arrivare alla formulazione della nozione di *Cognitive Language Engineering*. Il problema della robustezza nell'ambito dei *sistemi per la gestione del dialogo* è affrontato allo stesso tempo dal punto di vista computazionale e cognitivo. Verranno dunque proposti due modelli per la rappresentazione semantica dell'informazione e la sua conseguente assimilazione nello stato informazionale del dialogo. Il primo modello proposto permette di rappresentare gli

aspetti dinamici degli oggetti e degli eventi sui quali è possibile effettuare il ragionamento computazionale. Il secondo modello è incentrato sulla nozione di *spazio mentale* ed è utilizzato per costruire le rappresentazioni dei processi cognitivi legati al trattamento dell'informazione durante la comunicazione.

# Contents

<b>1. Introduction</b>	<b>1</b>
<b>I. Robust Language Engineering</b>	<b>11</b>
<b>2. Robust behaviour in Natural Language Systems</b>	<b>13</b>
2.1. Introduction . . . . .	13
2.1.1. Motivations and Goals . . . . .	14
2.1.2. Empiric evidence for Brittleness . . . . .	15
2.1.3. Problems . . . . .	15
2.1.4. Defining Robustness . . . . .	16
2.1.5. Robustness at different Linguistic Levels . . . . .	17
2.1.6. Robust Linguistics vs Robust NLP . . . . .	22
2.1.7. Statistical approaches to robust parsing . . . . .	23
2.1.8. Robustness in Natural Language Semantics . . . . .	24
2.1.9. Robust Computational Pragmatics . . . . .	25
2.1.10. Evaluation . . . . .	26
2.2. Key Issues in Robustness . . . . .	27
2.2.1. Extending Coverage . . . . .	27
2.2.2. Improving Efficiency . . . . .	28
2.2.3. The disambiguation problem . . . . .	28
2.2.4. Approximate Reasoning . . . . .	28
2.2.5. Enhancement of linguistic theories . . . . .	28
2.3. Discussion . . . . .	29
<b>3. A Computational Logic Toolbox</b>	<b>33</b>
3.1. Introduction . . . . .	33
3.2. Logic for Problem Solving . . . . .	34
3.2.1. Logic Programming . . . . .	36
3.2.2. Negation in Logic Programming . . . . .	38
3.3. Constraint Logic Programming . . . . .	41

## Contents

3.3.1. Examples of Constraint Domains and applications . . . . .	51
3.3.2. Constraint Handling Rules . . . . .	55
3.3.3. Soft Constraints in Logic Programming . . . . .	59
3.4. Modular Logic Programming . . . . .	64
3.4.1. Meta-Logic . . . . .	64
3.4.2. Algebra of Logic Program Expressions . . . . .	66
3.4.3. Compositional Semantics for Logic Programs . . . . .	66
3.4.4. The retraction operator . . . . .	71
3.5. Abductive Reasoning . . . . .	72
3.5.1. Abductive Logic Programming . . . . .	73
3.5.2. Application to Partial Order Planning with Event Calculus . . . . .	76
3.6. Summary . . . . .	79
<b>4. Computational Logic for Robust Language Engineering</b>	<b>81</b>
4.1. Introduction . . . . .	81
4.2. Logic for Natural Language Processing . . . . .	82
4.2.1. Definite Clause Grammars . . . . .	82
4.2.2. Extraposition Grammars . . . . .	84
4.3. Hybrid Approaches to Deductive Parsing . . . . .	91
4.3.1. Fuzzy CFG . . . . .	91
4.3.2. Probabilistic DCGs . . . . .	96
4.4. Feature Structures . . . . .	99
4.4.1. Feature Unification in Prolog . . . . .	101
4.5. Summary . . . . .	102
<b>5. Robustness “in the small”</b>	<b>105</b>
5.1. Introduction . . . . .	105
5.1.1. Applied Semantics for Natural Language Interfaces to Databases . . . . .	105
5.1.2. Computational logic for robust semantic analysis . . . . .	106
5.2. Robust Parsing in LHIP . . . . .	107
5.2.1. Island Parsing . . . . .	107
5.2.2. The LHIP parser compiler . . . . .	107
5.2.3. Using LHIP . . . . .	110
5.2.4. Extensions to LHIP: the ROTA project . . . . .	111
5.3. Interaction Through Speech with Information Systems . . . . .	114
5.4. The ISIS architecture . . . . .	115
5.4.1. Problem specification . . . . .	115

5.4.2. Query analysis . . . . .	116
5.5. Robust Semantic Analysis . . . . .	120
5.5.1. The ISIS robust semantic grammar . . . . .	122
5.5.2. Evaluation . . . . .	126
5.5.3. Discussion . . . . .	126
<b>6. Perspectives in Robustness “in the large”</b>	<b>129</b>
6.1. Introduction . . . . .	129
6.1.1. Modularity in Computational Linguistics . . . . .	131
6.1.2. From robust parsing to robust analysis . . . . .	132
6.1.3. Towards a Robust Natural Understanding System for Spoken Language . . . . .	132
6.2. Language Engineering Architectures . . . . .	133
6.2.1. Distributed Language Engineering . . . . .	134
6.2.2. Cooperative Language Engineering Architectures . . . . .	135
6.2.3. Discussion . . . . .	137
6.3. Agent-Oriented Language Engineering . . . . .	137
6.3.1. Agent Programming Models . . . . .	138
6.3.2. Agent Communication Languages . . . . .	142
6.3.3. Capability Brokering . . . . .	143
6.3.4. AOLE Software Layers . . . . .	145
6.4. Formal Methods for Multi-Agent Systems . . . . .	145
6.5. The HERALD architecture . . . . .	148
6.5.1. Language Engineering in HERALD . . . . .	149
6.5.2. A multi-agent view of robust NLP . . . . .	150
6.6. GETARUN for Robust Spoken Language Interpretation . . . . .	151
6.6.1. The GETARUN Architecture . . . . .	151
6.6.2. Objectives . . . . .	152
6.6.3. Software environment and tools . . . . .	153
6.6.4. Design Issues . . . . .	154
6.6.5. Modularity in GETARUN . . . . .	155
6.7. Discussion . . . . .	159
<b>II. Cognitive Language Engineering</b>	<b>161</b>

<b>7. Dialogue Systems</b>	<b>163</b>
7.1. Introduction . . . . .	163
7.1.1. What is Dialogue? . . . . .	163
7.1.2. Dialogue control layers . . . . .	165
7.1.3. States, events, causes . . . . .	165
7.1.4. Actions, agents, plans . . . . .	168
7.1.5. Mental Structures of Action . . . . .	170
7.1.6. Rational Agents . . . . .	171
7.1.7. Social Interaction . . . . .	173
7.1.8. Communicative acts . . . . .	174
7.1.9. Dialogue Acts . . . . .	176
7.2. Dialogue Systems . . . . .	177
7.2.1. Dialogue Management Systems . . . . .	178
7.2.2. Information State approach to Dialogue Management . . . . .	178
7.3. The representation of mental attitudes: the ViewGen system . . . . .	182
7.3.1. Attitudes report in dialogue with ViewGen . . . . .	185
7.3.2. Plan recognition in ViewGen . . . . .	187
7.3.3. Discussion . . . . .	188
<b>8. Knowledge Representation and Assimilation</b>	<b>189</b>
8.1. Introduction . . . . .	189
8.2. Computational Semantics and Computational Logic . . . . .	190
8.3. Situation Semantics . . . . .	191
8.3.1. Events and Processes in Situation Theory . . . . .	194
8.3.2. Inference in Situation Theory . . . . .	198
8.4. Logical Theories and Open Systems . . . . .	199
8.4.1. Situation Theory and Logic Programming . . . . .	200
8.4.2. Discussion . . . . .	202
8.5. From Semantic Interpretations to Dialogue Acts . . . . .	202
8.5.1. Discourse Models . . . . .	202
8.5.2. World models in Situation Semantics . . . . .	203
8.5.3. Semantic Interpretation in GETARUN . . . . .	205
8.5.4. Extraction of dialogue acts from semantic interpretation in GETARUN . . . . .	208
8.6. Summary . . . . .	208

<b>9. Reasoning with Objects and Actions</b>	<b>209</b>
9.1. Introduction . . . . .	209
9.2. Reasoning about Actions in multi-agent systems: Fluent Logic Programming . . .	211
9.2.1. Event Calculus . . . . .	211
9.2.2. Features & Fluents . . . . .	212
9.3. Fluent Logic Programming . . . . .	213
9.3.1. Fixpoint Semantics . . . . .	216
9.3.2. Meta-logical Semantics . . . . .	225
9.3.3. Soundness and Completeness . . . . .	228
9.4. Abductive Fluent Logic Programming . . . . .	229
9.5. Discussion . . . . .	233
9.5.1. Future works . . . . .	234
9.5.2. FLP and Situations Theory . . . . .	235
9.6. Summary . . . . .	235
<b>10. Reasoning with mental representations: ViewFinder</b>	<b>237</b>
10.1. Introduction . . . . .	237
10.2. Feature-based Topic Model . . . . .	239
10.2.1. Topics and Stereotypes . . . . .	240
10.2.2. Topic definition . . . . .	240
10.2.3. Feature Constraints . . . . .	241
10.2.4. Representing opinions . . . . .	243
10.2.5. Undefined features . . . . .	243
10.2.6. Opinion consistency . . . . .	244
10.3. Competency in believing . . . . .	245
10.4. Topic and Opinion viewpoints . . . . .	246
10.5. Environments . . . . .	247
10.5.1. The ViewFinder framework . . . . .	247
10.5.2. Ascription . . . . .	248
10.5.3. Adoption . . . . .	250
10.6. Feature-based Ascription and Adoption . . . . .	250
10.6.1. Representation of Nested environments . . . . .	251
10.6.2. Ascription . . . . .	252
10.6.3. Adoption . . . . .	253
10.6.4. Policies . . . . .	253
10.6.5. Interaction between Adoption and Ascription . . . . .	254
10.7. Querying the knowledge base . . . . .	254

*Contents*

10.7.1. Query on topic spaces . . . . .	255
10.8. Assimilating mental attitudes from communication . . . . .	256
10.8.1. Communication content language . . . . .	256
10.8.2. Knowledge assimilation . . . . .	258
10.9. Revising knowledge . . . . .	263
10.9.1. The “Untell” message . . . . .	263
10.10. Three Wise Men Puzzle revisited . . . . .	264
10.10.1. Intuitive solution . . . . .	264
10.10.2. Our approach . . . . .	265
10.10.3. Related works . . . . .	270
10.10.4. Common Knowledge: is it really necessary? . . . . .	272
10.11. Intensional objects and concept hierarchies . . . . .	273
10.12. Mental State Recognition from Communication . . . . .	273
10.12.1. Abductive Plan Recognition in ViewFinder . . . . .	274
<b>11. Conclusions</b>	<b>275</b>
11.1. The road towards Cognitive Language Engineering . . . . .	275
11.2. A BDI view of a Dialogue Move Engine . . . . .	277
11.3. Learning . . . . .	278
11.4. Applications: present and future . . . . .	279
11.4.1. Multi-modal dialogue interfaces . . . . .	280
11.4.2. Dialogue Annotation . . . . .	281
11.4.3. Intelligent Tutoring Systems . . . . .	284
11.5. Last but not least... . . . .	288