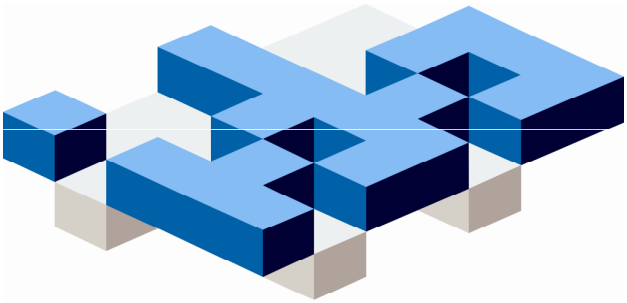
SWEDISH ARTIFICIAL INTELLIGENCE SOCIETY

The Swedish AI Society Workshop
May 27-28, 2009
IDA, Linköping University



DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE

LINKÖPINGS UNIVERSITET

Linköping University

ALDEBARAN Robotics

SAIS 2009

The Annual Swedish AI Society Workshop

May 27 – 28, 2009

Linköping, Sweden

Conference Proceedings

Organized by

Department of Computer and Information Science,
Linköping University

and

The Swedish AI Society

Edited by

Fredrik Heintz and Jonas Kvarnström

Published for

The Swedish AI Society

by Linköping University Electronic Press
Linköping, Sweden, 2009

Linköping Electronic Conference Proceedings, No. 35
Linköping University Electronic Press
Linköping, Sweden, 2009

ISSN 1650-3686 (print)
http://www.ep.liu.se/ecp/035/
ISSN 1650-3740 (online)

The SAIS Workshop 2009 Proceedings

Welcome!

We are happy to welcome you to the 25th annual workshop of the Swedish Artificial Intelligence Society (SAIS). The SAIS workshop provides an ideal forum for contacts among AI researchers and practitioners in Sweden and neighboring countries, as well as for establishing links with related research disciplines and industry.

The purpose of the SAIS workshop is:

- To give PhD students an opportunity to present their research to a friendly and knowledgeable audience and receive valuable feedback.
- To provide a forum for established researchers and practitioners to present past and current research contributing to the state of the art of AI research and applications.
- To provide an informal social event where AI researchers and practitioners can meet.

In addition to the six regular presentations we are proud to present six impressive invited speakers.

- Wolfram Burgard, Albert Ludwigs Universität Freiburg
- Patric Jensfelt, KTH
- Danica Kragic, KTH
- Magnus Merkel, Linköping University / Fodina Language Technology AB
- Christian Schulte, KTH
- Tom Ziemke, Skövde University

During the workshop Aldebaran Robotics will demonstrate their humanoid robot Nao and the winner of the SAIS Master's Thesis Award Robert Johansson, Örebro University, will present his work.

This year the yearly SweConsNet workshop is colocated with the SAIS Workshop. There will be one common invited talk and one common session which we hope will create mutual interest in the related areas.

Enjoy the workshop!
The SAIS 2009 Organizers

Committees

Organizing Committee

- Patrick Doherty, Linköping University
- Fredrik Heintz (chair), Linköping University
- Jonas Kvarnström (co-chair), Linköping University

Program Committee

- Christian Balkenius, Lund University
- Marcus Bjäreland, AstraZeneca
- Henrik Boström, University of Skövde
- Mathias Broxvall, Örebro University
- Paul Davidsson, Blekinge Institute of Technology
- Patrick Doherty, Linköping University
- Patrik Eklund, Umeå University
- Göran Falkman, University of Skövde
- Pierre Flener, Uppsala University
- Peter Funk, Mälardalen University
- Fredrik Heintz, Linköping University
- Anders Holst, SICS
- Kai Hübner, KTH
- Sture Hägglund, Linköping University
- Arne Jönsson, Linköping University
- Lars Karlsson, Örebro University
- Jonas Kvarnström, Linköping University
- Jan Eric Larsson, Lund University
- Martin Magnusson, Linköping University
- Jacek Malec, Lund University
- Michael Minock, Umeå University
- Lars Mollberg, Ericsson
- Anthony Morse, University of Skövde
- Mattias Nyberg, Linköping University and Scania AB
- Thorsteinn Rögnvaldsson, University of Halmstad and University of Örebro
- Lambert Spaanenburg, Lund University
- Cecilia Sönströd, University of Borås
- Vivian Vimarlund, Linköping University
- Ning Xiong, Mälardalen University

Program

Wednesday May 27

08:30-09:00 Registration

09:00-09:05 Opening

09:05-10:05 Invited talk

- Christian Schulte, KTH
  Constraint Programming for Real

10:05-10:30 Coffee break

10:30-11:30 Session 1

- Patrick Lambrix, Qiang Liu and He Tan
  Aligning Anatomy Ontologies in the Ontology Alignment Evaluation Initiative
- Michael Minock
  Discernability and Preference in Interactive Option Searches

11:30-12:30 Lunch, Ljusgården

12:30-13:45 Common session with SweConsNet

13:45-13:55 Quick break

13:55-15:00 Invited talks

- Danica Kragic, KTH
  Vision for Object Manipulation and Grasping
- Patric Jensfelt, KTH
  Spatial Modeling for Cognitive Systems

15:00-15:15 Quick introduction of posters

15:15-15:45 Coffee break with posters

15:45-16:45 SAIS Yearly Meeting

16:45-17:00 Refreshments

17:00-17:30 Presentation of Nao humanoid robot

17:30-18:30 Demonstration of the Nao

18:30-xx:xx Conference dinner

Thursday May 28

09:00-10:00 Invited talk

- Wolfram Burgard, Albert Ludwigs Universität Freiburg
  Probabilistic Techniques for Mobile Robot Navigation

10:00-10:30 Coffee break with posters

10:30-11:20 Session 2

- Alberto Montebelli, Robert Lowe and Tom Ziemke
  Embodied Anticipation in Neurocomputational Cognitive Architecture
- Ning Xiong and Peter Funk
  Integrating Case-Based Inference and Approximate Reasoning for Decision Making under Uncertainty

11:20-11:30 Quick break

11:30-12:00 SAIS Master's Thesis Award

- Robert Johansson, Örebro University
  Navigation on an RFID Floor

12:00-13:00 Lunch, Ljusgården

13:00-13:30 Invited talk

- Magnus Merkel, Linköping University / Fodina Language Technology
  Large-Scale Bilingual Extraction and Validation of Structured Patent Terminology

13:30-14:20 Session 3

- Martin Magnusson and Patrick Doherty
  Planning Speech Acts in a Logic of Action and Change
- Jacek Malec
  Active logic and practice

14:20-14:40 Coffee break

14:40-15:10 Invited talk

- Tom Ziemke, Skövde University
  Why Robots Need Emotions

15:10-15:20 Closing

Participants

- Mobyen Uddin Ahmed, Mälardalen University
- Marcus Bjäreland, AstraZeneca R&D Mölndal
- Wolfram Burgard, Albert Ludwigs Universität Freiburg
- Baki Cakici, Swedish Institute for Infectious Disease Control
- Gianpaolo Conte, IDA, Linköpings universitet
- Paul Davidsson, Blekinge Tekniska Högskola
- Patrick Doherty, IDA, Linköpings universitet
- Wlodek Drabent, IDA, LiU & IPI PAN Warszawa
- Peter Funk, IDT, Mälardalen university
- Tommy Färnqvist, IDA, Linköpings universitet
- David Hall
- Jun He, Department of IT, Uppsala University
- Fredrik Heintz, IDA, Linköpings universitet
- Anders Holst, SICS
- Sture Hägglund, IDA, Linköpings universitet
- Patric Jensfelt, Centre for Autonomous Systems, KTH
- Peter Jonsson, IDA, Linköpings universitet
- Arne Jönsson, IDA, Linköpings universitet
- Lars Karlsson, AASS, Örebro
- Håkan Kjellerstrand
- Danica Kragic, KTH
- Krzysztof Kuchcinski, Lunds Universitet
- Fredrik Kuivinen, IDA, Linköpings universitet
- Jonas Kvarnström, IDA, Linköpings universitet
- Mikael Zayenz Lagerkvist, KTH
- Patrick Lambrix, IDA, Linköpings universitet
- David Landén, IDA, Linköpings universitet
- Andreas Launila, KTH
- Niklas Lavesson, Blekinge Institute of Technology
- Tomas Lidén, Jeppesen Systems AB
- Qiang Liu, IDA, Linköpings universitet
- Martin Magnusson, IDA, Linköpings universitet
- Jacek Malec, Dept. of Computer Science, Lund University
- Magnus Merkel, IDA, Linköpings universitet
- Michael Minock, Umeå University
- Alberto Montebelli, University of Skövde, School of Humanities and Informatics

- Gustav Nordh, IDA, Linköpings universitet
- Per-Magnus Olsson, IDA, Linköpings universitet
- Umut Orguner, ISY, Linköpings universitet
- Justin Pearson, Department of IT, Uppsala University
- Federico Pecora, Örebro Universitet
- Jose Manuel Peña, IDA, Linköpings universitet
- Tommy Persson, IDA, Linköpings universitet
- Bruno Petit, Aldebaran Robotics
- Carl Christian Rolf, Lund University
- Chandan Roy, IDA, Linköpings universitet
- Piotr Rudol, IDA, Linköpings universitet
- Mohammad Saifullah, IDA, Linköpings universitet
- Christian Schulte, KTH
- Thomas Schön, Division of Automatic Control, Linköping University
- Kristoffer Sjöö, KTH
- Per Skoglar, ISY, Linköpings universitet
- Cecilia Sönström, School of Business and Informatics, University of Borås
- Anders Tunevi, Telia Sonera
- David Törnqvist, ISY, Linköpings universitet
- Håkan Warnquist, IDA, Linköpings universitet / Scania
- Vivian Vimarlund, IDA, Linköpings universitet
- Mariusz Wzorek, IDA, Linköpings universitet
- Ning Xiong, Mälardalen University
- Tom Ziemke, University of Skövde
- Magnus Ågren, SICS

Probabilistic Techniques for Mobile Robot Navigation

Wolfram Burgard
Albert Ludwigs Universität Freiburg, Germany

In recent years, probabilistic techniques have enabled novel and innovative solutions to some of the most important problems in mobile robotics. Major challenges in the context of probabilistic algorithms for mobile robot navigation lie in the questions of how to deal with highly complex state estimation problems and how to control the robot so that it efficiently carries out its task. In this talk I will discuss both aspects and present an efficient probabilistic approach to solve the simultaneous mapping and localization problem for mobile robots. I will also describe how this approach can be combined with an exploration strategy that simultaneously takes into account the uncertainty in the pose of the robot and in the map. For all algorithms I will present experimental results, which have been obtained with mobile robots in real-world environments as well as in simulation. I will conclude the presentation with a discussion of open issues and potential directions for future research.

Large-Scale Bilingual Extraction and Validation of Structured Patent Terminology

Magnus Merkel
Linköping University / Fodina Language Technology AB

A leading IT company in the world expresses its goals as making all information available to everybody, anywhere and anytime. One of the obstacles to achieving this goal has to do with language and language barriers. Automated translation (or machine translation MT) has been a field of study since the fifties within AI and has been seen as the holy grail of language technology for almost as long. MT is a key component for the aforementioned company if they are to succeed in bringing information across language barriers.

In reality, there are two major directions in MT today. One is data-driven and focused on statistical processing of documents. The other is more traditional, and based on rule-based translation systems, where linguistic knowledge is encoded in large lexicons and grammar rules. The data-driven camp is convinced that more data will solve the problems, e.g. by feeding a statistical MT system with tens of millions of parallel sentences (original and the corresponding translations) the statistical machinery will be able to create language and translation models that will produce high-quality translations. The rule-based MT camp believes that there are inherent features of human language that can never be modelled by massive amounts of data, and furthermore, that there simply are not enough parallel data for many language pairs to be found.

One subject field where high-quality automatic translations would be extremely useful concerns the patent area. Patent information is crucial for many businesses and to obtain patents and protect products are costly, for many reasons. The European Patent Office (EPO) has launched an automatic translation service on the Internet where patent agents can search approved patent applications and have them translated into several languages. This service is intended to be expanded to cover all European languages at the end of the project.

In this talk I will describe a large-scale extraction project of patent terminology (English-Swedish), which will be plugged into the EPO web service during the summer of 2009. The MT architecture on which the service is built is a rule-based architecture, where English is used as a pivot language and modules for translating

between any language X and English is being built. Starting in the summer of 2009, patent terminology was extracted from a set of English-Swedish document pairs. Information on the correct linguistic inflection patterns and hierarchical partitioning of terms based on their use was of utmost importance.

The process contains six phases, 1) Automatic analysis of the source material and system configuration; 2) Automatic term candidate extraction; 3) Term candidate filtering and initial linguistic validation; 4) Manual validation by domain experts; 5) Final linguistic validation; and 6) Publishing the validated terms.

Input to the extraction process consisted of more than 91.000 patent document pairs in English and Swedish, 565 million words in English and 450 million words in Swedish. The English documents were supplied in EBD SGML format and the Swedish documents were supplied in OCR processed scans of patent documents. After grammatical and statistical analysis, the documents were word aligned. Using the word-aligned material, candidate terms were extracted based on linguistic patterns. 750,000 term candidates were extracted and stored in a relational database. The term candidates were processed in 8 months resulting in 181.000 unique validated term pairs, which were then exported into several hierarchically organized terminology files, to be plugged into the rule-based MT system.

Constraint Programming for Real

Christian Schulte
Royal Institute of Technology

Since the inception of constraints in AI for modeling and solving combinatorial problems in the 1960's, constraint programming (CP) has emerged both as a scientific field as well as an array of successful techniques and tools for solving difficult real-life problems. Its applications are ubiquitous and include configuration, design, computational biology, diagnosis, logistics, planning, routing, and scheduling. The progress of CP is due to its multidisciplinary nature which includes fields such as AI, programming languages and systems, logics, operations research, and algorithmics.

In this talk, I attempt to give you the basic setup of CP for solving real-life combinatorial optimization problems. I will take you on several gentle excursions that shed light on the what, why, and how of CP for real: capturing structure in combinatorial problems by constraints that ease modeling and aid solving and CP as an amazingly flexible and powerful toolbox of constraints as reusable software components. During that journey, I will relate CP to other techniques such as SAT, linear programming, and the original model of constraint satisfaction that emerged from AI and point out its strengths and weaknesses.

Aligning Anatomy Ontologies in the Ontology Alignment Evaluation Initiative

Patrick Lambrix, Qiang Liu, He Tan
Department of Computer and Information Science
Linköpings universitet
581 83 Linköping, Sweden

Abstract

In recent years many ontologies have been developed and many of these ontologies contain overlapping information. To be able to use multiple ontologies they have to be aligned. In this paper we present and discuss results from aligning ontologies in a real case, the anatomy case in the 2008 Ontology Alignment Evaluation Initiative. We do this by briefly describing a base system for ontology alignment, SAMBO, and an extension, SAMBOdtf, and present and discuss their results for the anatomy case. SAMBO and SAMBOdtf performed best and second best among the 9 participating systems. SAMBO uses a combination of string matching and the use of domain knowledge. SAMBOdtf uses the same strategies but, in addition, uses an advanced filtering technique that augments recall while maintaining a high precision. Further, we describe the first results on ontology alignment using a partial reference alignment.[1]

1 Introduction

In recent years many ontologies have been developed. Intuitively, ontologies (e.g. [6]) can be seen as defining the basic terms and relations of a domain of interest, as well as the rules for combining these terms and relations. They are considered to be an important technology for the Semantic Web. Ontologies are used for communication between people and organizations by providing a common terminology over a domain. They provide the basis for interoperability between systems. They can be used for making the content in information sources explicit and serve as an index to a repository of information. Further, they can be used as a basis for integration of information sources and as a query model for information sources. They also support clearly separating domain knowledge from application-based knowledge as well as validation of data sources. The benefits of using ontologies include reuse, sharing and portability of knowledge across platforms, and improved documentation, maintenance, and reliability (e.g. [7]). Ontologies lead to a better understanding of a field and to more effective and efficient handling of information in that field.

Many of the currently developed ontologies contain overlapping information. For instance, Open Biomedical Ontologies (OBO, http://www.obofoundry.org/) lists 26 different anatomy ontologies (January 2009). Often we would want to be able to use multiple ontologies. For instance, companies may want to use community standard ontologies and use them together with company-specific ontologies. Applications may need to use ontologies from different areas or from different views on one area. Ontology builders may want to use already existing ontologies as the basis for the creation of new ontologies by extending the existing ontologies or by combining knowledge from different smaller ontologies. In each of these cases it is important to know the relationships between the terms in the different ontologies. Further, the data in different data sources in the same domain may have been annotated with different but similar ontologies. Knowledge of the inter-ontology relationships would in this case lead to improvements in search, integration and analysis of data. It has been realized that this is a major issue and some organizations have started to deal with it.

[1] This paper is partly a revised and updated version of the paper [12] focusing on the anatomy ontology alignment task, and partly an extended version. The former paper contains brief descriptions of the systems, but, following the tradition of the Ontology Alignment Evaluation Initiative, was written before the final results were available.

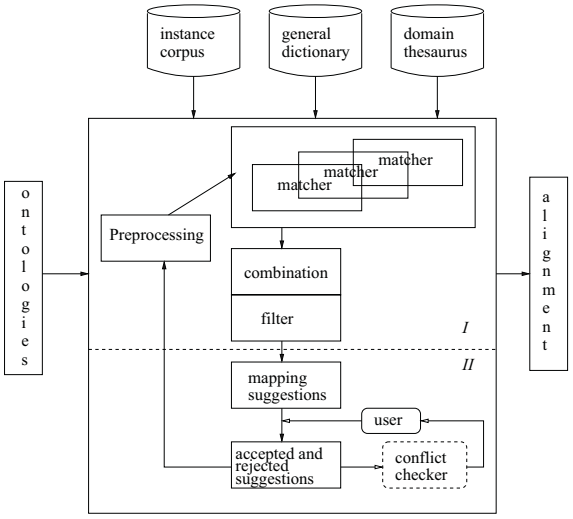


Figure 1: Alignment framework [11, 10].

In the remainder of this paper we say that we align two ontologies when we define the mapping relationships between terms in the different ontologies. We discuss results from aligning ontologies in a real case, the anatomy case in the 2008 Ontology Alignment Evaluation Initiative, one of the best known benchmark cases. We do this by presenting the two state-of-the-art ontology alignment systems that performed best and second best, and present and discuss their results for the anatomy case.

2 Background

2.1 Framework

A large number of ontology alignment systems have been developed. For an overview of most of these systems, we refer to review papers (e.g. [11, 16, 15, 8]), the ontology matching book [4], and the ontology matching web site at http://www.ontologymatching.org/.

Many ontology alignment systems are based on the computation of similarity values between terms in different ontologies and can be described as instantiations of the general framework defined in [11, 10] (figure 1). The framework consists of two parts. The first part (*I* in figure 1) computes mapping suggestions. The second part (*II*) interacts with the user to decide on the final mappings.

An alignment system receives as input two source ontologies. The ontologies can be preprocessed, for instance, to select pieces of the ontologies that are likely to contain matching terms. The alignment algorithm includes one or several matchers, which calculate similarity values between the terms from the different source ontologies and can be based on knowledge about the linguistic elements, structure, constraints and instances of the ontology. Also auxiliary information can be used. Mapping suggestions are then determined by combining and filtering the results generated by one or more matchers. By using different matchers and combining and filtering the results in different ways we obtain different alignment strategies. The suggestions are then presented to the user who accepts or rejects them. The acceptance and rejection of a suggestion may influence further suggestions. Further, a conflict checker is used to avoid conflicts introduced by the mappings. The output of the ontology alignment system is an alignment which is a set of mappings between terms from the source ontologies.

2.2 SAMBO and SAMBOdtf

SAMBO and SAMBOdtf are based on the framework described in section 2.1. They do not have a preprocessing step. SAMBO and SAMBOdtf contain currently five basic matchers [11]: two terminological matchers (a basic matcher and an extension using WordNet; extension described below), a structure-based matcher (which uses the is-a and part-of hierarchies of the source ontologies), a matcher based on domain knowledge (described below), and a learning matcher (which uses life science literature that is related to the concepts in the ontologies to define a similarity value between the concepts). In addition to these techniques we have also experimented with other matchers [13, 18, 21].

The user is given the choice to employ one or several matchers during the alignment process. We have two strategies to combine the results from different matchers. One strategy is to give weights to the different matchers and the similarity values are then computed as a weighted sum of the similarity values computed by the different matchers. The other strategy defines the similarity of a pair of terms as the maximum value of the similarity values for the pair computed by the different matchers.

The filtering method in SAMBO is single threshold filtering. Pairs of terms with a similarity value higher than or equal to a given threshold value are returned as

mapping suggestions to the user.

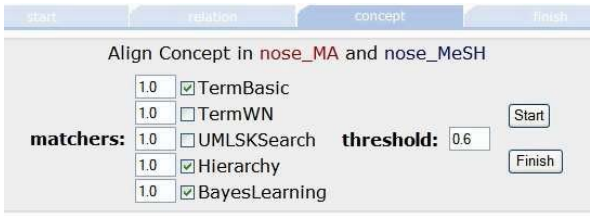


Figure 2: Combination and filtering.

Figure 2 shows a screenshot from the SAMBO system with the five matchers, a weighted sum combination and the single threshold filtering.

SAMBOdtf implements the double threshold filtering method developed in [2]. The double threshold filtering approach uses the structure of the ontologies. It is based on the observation that (for the different approaches in the evaluation in [11]) for single threshold filtering the precision of the results decreases and the recall increases when the threshold decreases. Therefore, we propose to use two thresholds. Pairs with similarity value equal to or higher than the upper threshold are retained as suggestions. The intuition is that this gives suggestions with a high precision. Further, pairs with similarity values between the lower and the upper threshold are filtered using structural information and the rest is discarded. We require that the pairs with similarity values between the two thresholds are 'reasonable' from a structural point of view.[2] The intuition here is that the recall is augmented by adding new suggestions, while at the same time the precision stays high because only structurally reasonable suggestions are added. The double threshold filtering approach contains the following three steps. (i) Find a consistent suggestion group from the pairs with similarity value equal to or higher than the upper threshold. We say that a set of suggestions is a consistent suggestion group if each concept occurs at most once as first argument in a pair, at most once as second argument in a pair and for each pair of suggestions (A,A') and (B,B') where A and B are concepts in the first ontology and A' and B' are concepts in the second ontology: $A \subset B$ iff $A' \subset B'$. (ii) Use the consistent suggestion group to partition the original ontologies. (iii) Filter the pairs with similarity values between the lower and upper thresholds using the partitions. Only pairs of which the elements belong to corresponding pieces in the partitions are retained as suggestions. For details we refer to [2].

[2] In our implementation we have focused on the is-a relation.

Figure 3: Mapping suggestion.

Based on the results from the matchers, combination and filtering algorithms, mapping suggestions are provided to the user. Figure 3 shows such a suggestion. SAMBO displays information (definition/identifier, synonyms, relations) about the source ontology terms in the suggestion. For each mapping suggestion the user can decide whether the terms are equivalent, whether there is an is-a relation between the terms, or whether the suggestion should be rejected. If the user decides that the terms are equivalent, a new name for the term can be given as well. Upon an action of the user, the suggestion list is updated. If the user rejects a suggestion where two different terms have the same name, she is required to rename at least one of the terms. The user can also add comments on a mapping relationship. At each point in time during the alignment process the user can view the ontologies represented in trees with the information on which actions have been performed, and she can check how many suggestions still need to be processed. A similar list can be obtained to view the previously accepted mapping suggestions. In addition to the suggestion mode, the system also has a manual mode in which the user can view the ontologies and manually map terms.

2.3 Ontology Alignment Evaluation Initiative - Anatomy case

The Ontology Alignment Evaluation Initiative (OAEI, http://oaei.ontologymatching.org/) is a yearly initiative that was started in 2004. The goals are, among others, to assess the strengths and weaknesses of alignment systems, to compare different techniques and to improve evaluation techniques. This is to be achieved through controlled experimental evaluation. For this purpose OAEI publishes different cases of ontology alignment problems, some of which are open (reference alignment

is known beforehand), but most are blind (reference alignment is not known - participants send their mapping suggestions to organizers who evaluate the performance).

In the anatomy case (version 2008) participants are required to align the Adult Mouse Anatomy (2744 concepts) and the NCI Thesaurus - anatomy (3304 concepts). The case is divided into 4 tasks (of which task 4 was new for 2008). The anatomy case is a blind case. The reference alignment (the correct solution according to the organizers) contains 1523 equivalence mappings of which 934 are deemed trivial (i.e. they can be found by a relatively basic string-based matcher). Only equivalence correspondences between concepts are considered.

In all tasks the two ontologies should be aligned. The results of the experiments are given in terms of the quality of the mapping suggestions. The evaluation measures are precision, recall, recall+ and f-measure. *Precision* measures how many of the mapping suggestions were correct. It is defined as the number of correct suggestions divided by the number of suggestions. *Recall* measures how many of the correct mappings are found by the alignment algorithm. It is defined as the number of correct suggestions divided by the number of correct mappings. *Recall+* is the recall computed with respect to non-trivial mappings. *F-measure* is the weighted harmonic mean of precision and recall.

In task 1 the system should be tuned to optimize the f-measure. This means that both precision and recall are important. The systems are compared with respect to precision, recall, f-measure and recall+. For the f-measure in task 1, precision and recall are evenly weighted. Nine systems participated in this task.

In tasks 2 and 3, in which four systems participated, the system should be optimized with respect to precision and recall, respectively. The f-measure is computed with an unevenly weighted precision and recall (factor 5).

In task 4, in which four systems participated, a partial reference alignment is given which can be used during the computation of mapping suggestions. It contains all trivial and 54 non-trivial mappings in the reference alignment. In this case precision, recall and f-measure are computed with respect to the non-given part of the reference alignment.

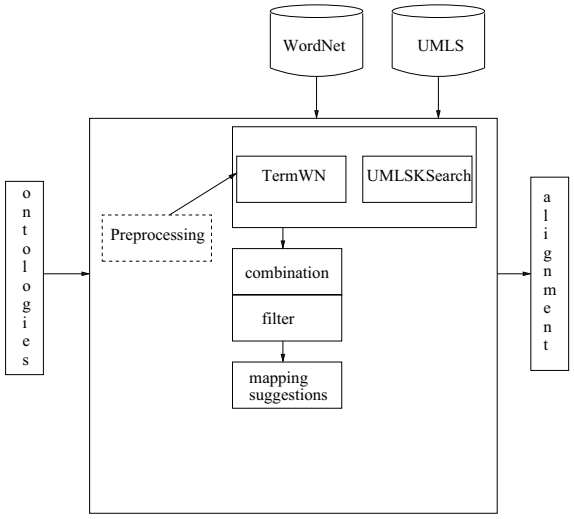


Figure 4: SAMBO and SAMBOdtf for OAEI.

3 SAMBO and SAMBOdtf for OAEI

The OAEI only evaluates the non-interactive part of the ontology alignment systems. Therefore, we used a variant of the systems without the user interface (see figure 4). Further, it would not make sense to have mapping suggestions where a concept appears more than once as the user would not be able to make a choice. Therefore, we decided to filter our systems' mapping suggestion lists such that only suggestions are retained where the similarity between the concepts in the mapping suggestion is higher than or equal to the similarity of these concepts to any other concept according to the mapping suggestion list. (In the case there are different possibilities, one is randomly chosen. In the implementation the first in the list is chosen.)

For the OAEI we used the following matchers. The matcher *TermWN* contains matching algorithms based on the textual descriptions (names and synonyms) of concepts and relations. In the current implementation, the matcher includes two approximate string matching algorithms (n-gram and edit distance), and a linguistic algorithm that also uses WordNet (http://wordnet.princeton.edu/) to find synonyms and is-a relations. Our matcher *UMLSKSearch* uses the Metathesaurus in the Unified Medical Language System (UMLS, http://www.nlm.nih.gov/research/umls/). The similarity of two terms in the source ontologies is

determined by their relationship in UMLS. In our experiments we used the UMLS Knowledge Source Server to query the UMLS Metathesaurus with source ontology terms. The querying is based on searching the normalized string index and normalized word index provided by the UMLS Knowledge Source Server. We used version 2008AA of UMLS. As a result we obtain concepts that have the source ontology term as their synonym. We assign a similarity value of 0.99 if the source ontology terms are synonyms of the same concept and 0 otherwise.

The combination algorithm used for OAEI 2008 is a maximum-based algorithm. The similarity value for a pair of concepts is the maximum value obtained from TermWN and UMLSKSearch for this pair of concepts.

As in the full SAMBO and SAMBOdtf systems, SAMBO uses single threshold filtering and SAMBOdtf double threshold filtering.

4 Results for the Anatomy Case

The results for the participating systems and discussions are available from http://oaei.ontologymatching.org/2008/ and the paper [1].

For SAMBO and SAMBOdtf tests were performed on a IBM R61i Laptop, WinXP Intel(R) Pentium(R) Dual T2370 @ 1.73GHz, 1.73GHz, 1.99G RAM.

Task 1. We used the matchers, combinations and filtering described in section 3. SAMBO used threshold 0.6. SAMBOdtf used upper threshold 0.8 and lower threshold 0.4. These thresholds were chosen based on our experience with previous experiments with biomedical ontologies. SAMBO generated 1465 mapping suggestions and reached a precision of 0.869, a recall of 0.836 and an f-value of 0.852. Further, it reached a recall+ of 0.586. This was the best result for all 9 participating systems in OAEI 2008.[3] In 2007 we used a version of SAMBO that used Term instead of TermWN and a previous version of UMLS. The 2007 version obtained a better recall for non-trivial mappings, but at the cost of an overall decrease in precision and recall. A possible explanation for this is our strategy for choosing maximum one mapping suggestion per concept. In 2008 exact matching strings were preferred, while in 2007 there was no preference between pairs that had exact matching strings or pairs that were proposed based on domain knowledge.

SAMBOdtf generates 1527 mapping suggestions. Of these suggestions, 1440 have a similarity value between 0.6 and 0.8. This means that SAMBOdtf filtered out 25 of the suggestions obtained by SAMBO with threshold 0.6. (A manual check seems to suggest that most of these are correctly filtered out, but some are wrongly filtered out. One reason for removing correct suggestions is that the source ontologies have missing is-a links.) Further, SAMBOdtf also filtered out 19 suggestions with similarity values between 0.4 and 0.6. (A manual check seems to suggest that these were correctly filtered out.) SAMBOdtf obtained a precision of 0.831, a recall of 0.833, an f-value of 0.832 and a recall+ of 0.579. This was the second best result for all 9 participating systems in OAEI 2008.

The running time for SAMBO was ca 12 hours and for SAMBOdtf ca 17 hours. As discussed in [1] the best performing systems in 2007 and 2008 in terms of quality of the mapping suggestions heavily use domain knowledge. This comes at a cost of larger running time. However, another interesting observation discussed by the organizers of the anatomy track at OAEI 2008 was that circa 50% of the non-trivial mappings was found by at least one system using domain knowledge and at least one system that did not use domain knowledge. Circa 13% of the non-trivial mappings was found only by systems using domain knowledge. Circa 13% of the non-trivial mappings was found only by systems that did not use domain knowledge. The reason for this is that the used domain knowledge (most often UMLS) is not complete. Further, still circa 25% of the non-trivial mappings were not found at all. As [1] suggests, a combination of different strategies may improve the results. Taking the union of the SAMBO results with the results of the RiMOM [24] and Lily [22] systems would give a higher recall and recall+. RiMOM and Lily use linguistic and structure-based approaches, but no domain knowledge.

Tasks 2 and 3. We did not participate in tasks 2 and 3. As reported in [1] the best system for task 2 (RiMOM) obtained a precision of 0.964 (with a recall of 0.677). The best system for task 3 (RiMOM) obtained a recall of 0.808 (with a precision of 0.450 and a recall+ of 0.538). We note that the best recall is lower than the recall for SAMBO and SAMBOdtf in task 1. The best system for task 3 for non-trivial mappings (Lily) obtained a recall+ of 0.613 (with a recall of 0.790 and a

[3] The system with best f-measure in 2007 (AOAS [23]) obtained 0.928 precision, 0.815 recall, 0.523 recall+ and 0.868 f-measure. SAMBO (in its first participation) was second best system in 2007 regarding precision, recall and f-value, but best regarding recall+ [20].

precision of 0.490). As neither RiMOM nor Lily used domain knowledge, these can be considered to be good results.

Task 4. For task 4, we augmented SAMBO and SAMBOdtf in the following ways.

For SAMBO we added the mappings in the partial reference alignment to the list of mapping suggestions, but with a special status. These mappings could not be removed in any filtering step. SAMBO generated 1494 suggestions of which 988 are also in the partial reference alignment. SAMBO obtained the best results of the participating systems. With respect to the unknown part of the reference alignment, its precision increased with 0.024, its recall decreased with 0.002 and its f-value increased with 0.011. Our strategy for using the partial reference alignment helped remove wrong suggestions that conflicted with the partial reference alignment, although also some correct suggestions were removed.

For SAMBOdtf we also added the mappings in the partial reference alignment to the list of mapping suggestions with the special status. In addition, we used the partial reference alignment in the double threshold filtering step. We used a part of the partial reference alignment that satisfied the consistent group property as a consistent suggestion group. For upper threshold 0.8 and lower threshold 0.4 we obtained 1547 mapping suggestions. SAMBOdtf obtained the second best results of the participating systems. With respect to the unknown part of the reference alignment, its precision increased with 0.040, its recall with 0.008 and its f-value with 0.025. SAMBOdtf was the system with the highest increase in f-value and was the only system that used the partial reference alignment to increase both precision and recall. This result is most likely due to the fact that, in contrast to task 1 where the consistent suggestion group consists of suggestions, in this task the consistent suggestion group consists of true mappings. Therefore, the suggestions with similarity value between the two thresholds that are retained are structurally reasonable with respect to true mappings and not just (although with high confidence) suggestions.

We note that although the improvements seem small, as SAMBO and SAMBOdtf perform already well on their own, even small improvements are valuable. Further, due to the choice of the partial reference alignment all newly found mappings are non-trivial.

In a follow-up on task 4 we have started investigating the use of partial reference alignments in the different components of the framework in section 2.1 [10]. In addition to the techniques described above, we have used partial reference alignments in a preprocessing step, to define new matchers and in new filtering steps.[4]

In the preprocessing approaches we investigate whether we can use a partial reference alignment to partition the ontologies into mappable parts and test whether, in addition to the fact that we do not have to compute similarity values between all terms from the first ontology and all terms from the second ontology, this also leads to a better quality of the mapping suggestions. In the first approach we partition the ontologies into mappable parts using the partitioning step of the double threshold filtering described in section 2.2 and [2]. A part of the partial reference alignment satisfying the consistent group property is used as a consistent group. Further, according to our experience in aligning ontologies we know that the structure of the source ontologies is not always perfect. For instance, given the two ontologies and the partial reference alignment in the anatomy case of OAEI 2008, it can be deduced that many is-a relations are missing in at least one of the source ontologies. Based on this observation we experiment with a second approach where we add to the source ontologies the missing is-a relationships that can be deduced from the source ontologies and the partial reference alignment. After this 'fixing' of the source ontologies the partial reference alignment will satisfy the consistent group property. As the intuition of the preprocessing step is to partition the ontologies into mappable parts, we can only generate mapping suggestions that are reasonable from a structural point of view. This suggests that, when using a preprocessing step, the precision may become higher as suggestions that do not conform to the structure of the source ontologies cannot be made. As we add the partial reference alignment to the result, the recall may be increased as some of the partial reference alignment mappings may not be found by the base systems. However, the similarity values between the terms do not change and it is therefore not likely that new mappings are found. For thresholds 0.6 and above our experiments corroborate this intuition. Another observation is that, contrary to the intuition, fixing the source ontologies may lead to a decrease in recall. The reason for this is the quality of the underlying ontologies where 'is-a' is not always properly used.

One way to create a matcher based on a partial reference alignment, is to use underlying properties of

[4] Thanks to Christian Meilicke of the organization committee of OAEI Anatomy for running our newly developed algorithms on the anatomy data set.

the mappings in the partial reference alignment. We have previously observed that sometimes for two given source ontologies, common patterns can be found between the correct mappings. For instance, in the partial reference alignment of the OAEI 2008 anatomy we find the mappings <lumbar vertebra 5, l5 vertebra> and <thoracic vertebra 11, t11 vertebra> which share a similar linguistic pattern. Based on this observation we developed a matcher that augments previously generated similarity values for term pairs when these term pairs display a similar (linguistic) pattern as mappings in the partial reference alignment. Several new correct mappings were found.

Finally, we also experimented with a filter strategy that removes suggestions that do not have similar linguistic patterns than the mappings in the partial reference alignment. We expect therefore that some correct suggestions obtained through UMLS will be removed and therefore the recall may go down. This is indeed the case in our experiments. The precision when using this filter approach is, however, always higher or equal to the precision for SAMBO. This is because the suggestions that had a linguistically similar pattern as mappings in the partial reference alignment were usually correct.

5 Conclusion

We have briefly described our ontology alignment systems SAMBO and SAMBOdtf and their results for the anatomy alignment tasks of OAEI. We have used a combination of UMLSKSearch and TermWN and obtained the best results in OAEI anatomy 2008 with respect to quality of the suggestions. However, as the recall+ of the best system is still around 0.6, work still needs to be done to find non-trivial mappings.

Another problem that we investigate is whether systems that do well in the anatomy case will also perform well for other cases. More large-scale evaluation is needed in the area.

Further, the OAEI cases only provide a benchmark for part I of the framework described in section 2.1. Not so much work has been done on user involvement, user interfaces and ontology and ontology alignment visualization [9, 5].

Also, given the fact that different algorithms seem to do differently well for different kinds of ontologies and evaluation measures, a major problem is deciding which algorithms should be used for a given alignment task. This is a problem that users face, and that we have also faced in the evaluation. Recommendation strategies [19, 14, 3] may alleviate this problem.

Other challenges for the ontology alignment field are given in [17].

References

[1] C Caracciolo, J Euzenat, L Hollink, R Ichise, A Isaac, V Malaise, C Meilicke, J Pane, P Shvaiko, H Stuckenschmidt, O Svab-Zamazal, and V Svatek. Results of the ontology alignment evaluation initiative 2008. In *Proceedings of the Third International Workshop on Ontology Matching*, 2008.

[2] B Chen, H Tan, and P Lambrix. Structure-based filtering for ontology alignment. In *Proceedings of the IEEE WETICE Workshop on Semantic Technologies in Collaborative Applications*, pages 364–369, 2006.

[3] M Ehrig, S Staab, and Y Sure. Bootstrapping ontology alignment methods with APFEL. In *Proceedings of the International Semantic Web Conference*, pages 186–200, 2005.

[4] J Euzenat and P Shvaiko. *Ontology Matching*. Springer, 2007.

[5] S Falconer and M-A Storey. A cognitive support framework for ontology mapping. In *Proceedings of the 6th International Semantic Web Conference*, pages 114–127, 2007.

[6] A Gómez-Pérez. Ontological engineering: A state of the art. *Expert Update*, 2(3):33–43, 1999.

[7] R Jasper and M Uschold. A framework for understanding and classifying ontology applications. In *Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management*, 1999.

[8] Y Kalfoglou and M Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.

[9] P Lambrix and A Edberg. Evaluation of ontology merging tools in bioinformatics. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 589–600, 2003.

[10] P Lambrix and Q Liu. Using partial reference alignments to align ontologies. In *Proceedings of the 6th European Semantic Web Conference*, 2009.

[11] P Lambrix and H Tan. SAMBO - a system for aligning and merging biomedical ontologies. *Journal of Web Semantics, Special issue on Semantic Web for the Life Sciences*, 4(3):196–206, 2006.

[12] P Lambrix, H Tan, and Q Liu. SAMBO and SAMBOdtf results for the ontology alignment evaluation initiative 2008. In *Proceedings of the Third International Workshop on Ontology Matching*, pages 190–198, 2008.

[13] P Lambrix, H Tan, and W Xu. Literature-based alignment of ontologies. In *Proceedings of the Third International Workshop on Ontology Matching*, pages 219–223, 2008.

[14] M Mochol, A Jentzsch, and J Euzenat. Applying an analytic method for matching approach selection. In *Proceedings of the Workshop on Ontology Matching*, 2006.

[15] NF Noy. Semantic integration: A survey of ontology-based approaches. *Sigmod Record*, 33(4):65–70, 2004.

[16] P Shvaiko and J Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV:146–171, 2005.

[17] P Shvaiko and J Euzenat. Ten challenges for ontology matching. In *Proceedings of the 7th International Conference on Ontologies, Databases, and Applications of Semantics*, 2008.

[18] H Tan, V Jakonienė, P Lambrix, J Aberg, and N Shahmehri. Alignment of biomedical ontologies using life science literature. In *Proceedings of the International Workshop on Knowledge Discovery in Life Science Literature, LNBI 3886*, pages 1–17, 2006.

[19] H Tan and P Lambrix. A method for recommending ontology alignment strategies. In *Proceedings of the 6th International Semantic Web Conference, LNCS 4825*, pages 494–507, 2007.

[20] H Tan and P Lambrix. SAMBO results for the ontology alignment evaluation initiative 2007. In *Proceedings of the Second International Workshop on Ontology Matching*, pages 236–243, 2007.

[21] T Wächter, H Tan, A Wobst, P Lambrix, and M Schroeder. A corpus-driven approach for design, evolution and alignment of ontologies. In *Proceedings of the Winter Simulation Conference*, pages 1595–1602, 2006. Invited contribution.

[22] P Wang and B Xu. Lily: ontology alignment results for OAEI 2008. In *Proceedings of the Third International Workshop on Ontology Matching*, pages 167–175, 2008.

[23] S Zhang and O Bodenreider. Hybrid alginement strategy for anatomical ontologies: results of the 2007 ontology alignment contest. In *Proceedings of the Second International Workshop on Ontology Matching*, pages 139–149, 2007.

[24] X Zhang, Q Zhong, J Li, and J Tang. RiMOM results for OAEI 2008. In *Proceedings of the Third International Workshop on Ontology Matching*, pages 182–189, 2008.

Discernability and Preference in Interactive Option Searches

Michael Minock
Department of Computing Science
Ume University, Sweden 90187
mjm@cs.umu.se

Abstract

In option searches, a user seeks to locate an ideal option (e.g. a flight, restaurant, book, etc.) from a set of n such options. The aim of this paper is to provide a solid mathematical basis for optimizing presentation length in such searches. The paper develops an information theoretic model that takes into account the user's ability to discern among options as well as their *a priori* preference. The developed model makes definite predictions about what clusterings of a user query are more or less informative based on measures of *information gain*. Users are offered descriptions of such clusters as the basis for subsequent refinement steps in a drill-down dialogue to locate the best option. We have implemented an initial system that performs reasonably well on moderately large data sets and gives intuitively appealing results. The system is in the process of being integrated into a natural language interface system for end-user evaluation.

1 Introduction

As pointed out in [5], it is critical that spoken dialogue systems limit presentation duration for interactive option searches. Thus if the user requests "flights to Berlin leaving before noon," and there are many such flights, it is a mistake to simply start listing them in succession – the user would become irritated by the long descriptions and would be unlikely to remember enough detail to make an optimal choice. In the database of table 1 there are just four such flights, but even here it might be better to ask the follow up question, "Do you prefer Lufthansa or SAS?". Such *summarize-and-refine* (SR) techniques [4] cluster the options meeting the user's constraints into sets (e.g. "the SAS flights" and "the Lufthansa flights"), present these sets as summaries (or implicitly through questions) and then let the user refine the search to the cluster that interests them most. Such techniques promote efficiency by reducing what would be a linear number of descriptions to a roughly logarithmic number.

While such *summarize-and-refine* systems are particularly suited to spoken dialogue systems where users can reliably command systems to drill down into one or another summary, there are difficulties when such systems pick summaries that are not discernible to the user. For example if the system were to respond to the question above with "do you prefer flights on an A300 or an A320?", most users would be hard pressed to make an informed choice. The work presented here, inspired by [1], recasts the interactive search process in an information theoretic light and introduces a model of *discernibility* among options as well as a general parameter γ of intolerance for a sub-optimal results. The work's main contribution is to propose a more solid mathematical basis for optimizing presentation length in options searches.

2 Foundations

2.1 Options, databases, answer sets and clusterings

Consider the universe of values $\mathcal{U}$ and, for a given k, all the k-tuples $\mathcal{U}^k$, hereafter referred to as *options*. We denote the i-th value (starting at 1) of option t as $t[i]$. The set of *conditions* $\mathcal{C}$ are functions mapping $\mathcal{U}^k \rightarrow \{\mathsf{true}, \mathsf{false}\}$, that is for $c \in \mathcal{C}$ and option $t \in \mathcal{U}^k$, $c(t)$ is either true or false. Let $\mathcal{D}$ be a database of n options $t_1, .., t_n$. An *answer*

no.	dest	airline	dep	price	meal	aircraft
1	Paris	SAS	8	€200	yes	A300
2	Berlin	Luft	8	€250	yes	A320
3	London	SAS	9	€150	yes	A300
4	Paris	AF	9	€250	yes	A320
5	Berlin	Luft	9	€200	no	A320
6	London	BA	10	€200	yes	A320
7	Berlin	SAS	10	€250	no	A300
8	Berlin	SAS	11	€100	no	A300

Table 1: Example Last Minute Travel Database

set is denoted as $\{x|x \in \mathcal{D} \wedge Q(x)\}$ where $Q(x)$ is a boolean combination of conditions. Hereafter we will assume that $\mathcal{D}$ is fixed and thus drop explicit reference to it, instead describing answers sets as simply $\{x|Q(x)\}$. The semantics of answer sets are standard, where $(\forall t \in \mathcal{D})(t \in \{x|Q(x)\} \Leftrightarrow Q(t))$. Often we will refer to the expression $Q(x)$ as a *query*.

When deciding the next dialogue move after the user has identified $\{x|Q(x)\}$ as the set that they interested in, we must consider the possible clusterings $\langle Q(x) : Q_1(x), .., Q_m(x)\rangle$ which present m further summarize-and-refine sets to consider. As an example, the clustering of the query for "the flights to Berlin leaving before noon" into those on Lufthansa or SAS is:

$$\langle \{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x)\} : \{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x) \wedge \mathsf{onLuft}(x)\}, \{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(\mathsf{x}) \wedge \mathsf{onSAS}(x)\}\rangle$$

Note that our definition of a clustering puts no conditions on the relationship between $\{x|Q(x)\}$ and $\cup_{i=1}^{m}\{x|Q_i(x)\}$. Thus the relationship may be specialization, generalization or some combination thereof. For example a 'specializing' clustering of "the flights to Berlin leaving before noon" into those €100 euro or less is:

$$\langle \{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x)\} : \{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x) \wedge \mathsf{PriceLEQ}(x, 100)\}\rangle$$

A 'generalizing' clustering could be:

$$\langle \{x|\mathsf{beforeNoon}(x) \wedge \mathsf{toBerlin}(x)\} : \{x|\mathsf{before3PM}(x) \wedge \mathsf{toBerlin}(x) \wedge \mathsf{onLuft}(x)\}, \{x|\mathsf{before1PM}(x) \wedge \mathsf{toBerlin}(\mathsf{x}) \wedge \mathsf{onSAS}(x)\}\rangle$$

2.2 User preferences

A model of user preference captures the *a priori* assumptions about how the user values alternative options. Note that within a specific dialogue, users express hard conditions such as the destination or the need to fly at a specific time that are not captured in the user model. However given that a set of options meet the hard constraints supplied by the user, the user model will rank these options based on this *a priori* model. Moreover as we shall see below, based on notions of discernibility, options that fall outside of the user supplied hard constraints may in fact be worth presenting.

The work here allows for any type of quantitative model of user preference, but to avoid formal difficulties, we assume that for all $t \in \mathcal{D}$, $\mathsf{util}(t) > 0$. The following shorthand notation expresses total utility over an answer set:

$$\mathsf{util}(\{x|Q(x)\}) = \sum_{t \in \{x|Q(x)\}} \mathsf{util}(t)$$

2.3 Discernibility

In addition to the model of preference, there is a related model of discernibility. Two options are perfectly discernible if the user can immediately recognize them as being qualitatively different. For example, under a 'normal' context, a flight to Berlin versus a flight to Paris are perfectly discernible where as two flights to Berlin, one on an A300 and other on an A320 are not discernible.

Formally, for each i-th component of the options, assume that there is a function $\zeta_i : \mathcal{U} \times \mathcal{U} \rightarrow [0..1]$. The intuition of ζ_i is that if options t and t' agree on all components other than i (i.e. $t[j] = t'[j]$ for $1 \leq j \leq k$ and $j \neq i$), then $\zeta_i(t[i], t'[i])$ is the

probability that t and t' are indistinguishable to the user. The product of these measures gives an overall measure of similarity for tuples.

$$\mathsf{sim}(t,t') = \prod_{i=1}^{k} \zeta_i(t[i],t'[i])$$

Note that $\mathsf{sim}(t,t) = 1$ and that $\mathsf{sim}(t,t') = 0$ if there is at least one component upon which t and t' are perfectly discernible.

3 Our Approach

3.1 The ideal answer assumption

We make what we call the **ideal answer assumption** which states that there is some option $\mathsf{opt} \in \mathcal{D}$ which is the single best option that the user is searching for. The amount of information that can be usefully applied to locating opt is measured in bits, or answers to 'yes/no' questions. While in cases of perfect discernibility, it will take $\log_2 n$ bits to locate opt among n options, due to problems of discernibility only so many bits may be usefully employed to locate opt. Note that this is different from a measure of entropy. Consider the cases in which all options are completely indiscernible. Answering yes/no questions provides no information toward locating the ideal option. The best one can do in fact is simply pick one the options at random and present it as the ideal. Formally, we use the following definition of the information content within a cluster:

$$I(\{x|Q(x)\}) = \log_2\left(\frac{|\{x|Q(x)\}|^2}{\sum_{t'\in\{x|Q(x)\}}\sum_{\hat{t}\in\{x|Q(x)\}}\mathsf{sim}(t',\hat{t})}\right)$$

The prior probability of an option being ideal is proportional to its utility with respect to the model of *a priori* user preferences:

$$P(t=\mathsf{opt}) = \frac{\mathsf{util}(t)}{\mathsf{util}(\{x|x\in\mathcal{D}\})}$$

We introduce the notation $\mathsf{id}_{\hat{t}}$ to denote the situation in which the user has identified the option $\hat{t}$ as opt. Of course, based on problems of discernibility, the user could be wrong.

$$P(t=\mathsf{opt}|\mathsf{id}_{\hat{t}}) = \frac{\mathsf{sim}(t,\hat{t})}{\sum_{t'\in\{x|x\in\mathcal{D}\}}\mathsf{sim}(t',\hat{t})}$$

We now introduce the generalized notation id_Q to denote the situation in which the user has declared that $\mathsf{opt} \in \{x|Q(x)\}$. We obtain:

$$P(t=\mathsf{opt}|\mathsf{id}_Q) = \sum_{\hat{t}\in\{x|Q(x)\}} P(t=\mathsf{opt}|\mathsf{id}_{\hat{t}})\cdot\frac{\mathsf{util}(\hat{t})}{\mathsf{util}(\{x|Q(x)\})}$$

Note that we are weighing options in $\{x|Q(x)\}$ according to the model of user preference. This makes sense, because the model of preference gives us our *a priori* probability that a given option within $\{x|Q(x)\}$ would be selected as ideal by the user. Now we develop the full generalized form:

$$P(\mathsf{opt}\in\{x|Q'(x)\}|\mathsf{id}_Q) = \sum_{t'\in\{x|Q'(x)\}} P(t'=\mathsf{opt}|\mathsf{id}_Q)$$

3.2 Information Gain

The natural question to consider is how much information is gained through a clustering $\langle Q(x) : Q_1(x),..,Q_m(x)\rangle$. This is decided in the normal way by subtracting the information required to locate opt within $\{x|Q(x)\}$ from the information required to locate opt within each cluster $\{x|Q_i(x)\}$ weighted by the probability that the user will select the given cluster $Q_i(x)$ on their next refinement move. Finally consideration must be given to the possibility that the user refines the wrong cluster or that opt is not within any cluster $\{x|Q_i(x)\}$. In such a case the user suffers the cost γ measured in terms of bits. Given these ideas we arrive at the following measure of gain:

$$\begin{aligned}&\mathsf{gain}(\langle Q(x) : Q_1(x),..,Q_m(x)\rangle) = \\ &\quad I(\{x|Q(x)\}) + P(\mathsf{opt}\notin\{x|Q(x)\}|\mathsf{id}_Q)\cdot\gamma \\ &\quad - \\ &\quad \sum_{i=1}^{m} P(\mathsf{sel}_{Q_i})\cdot(I(\{x|Q_i(x)\}) + \\ &\qquad\qquad P(\mathsf{opt}\notin\{x|Q_i(x)\}|\mathsf{id}_{Q_i})\cdot\gamma) \\ &\quad - P(\mathsf{opt}\in\{x|Q(x)\bigwedge_{i=1}^{m}\neg Q_i(x)\}|\mathsf{id}_Q)\cdot\gamma\end{aligned}$$

where sel_{Q_i} means that the user will select Q_i as the basic of further refinement.

Using the model of user preferences we assume[1] that:

$$P(\mathsf{sel}_{Q_i}) \quad = \quad \frac{\mathsf{util}(\{x|Q_i(x)\})}{\sum_{Q' \in \{Q_1,..,Q_m\}} \mathsf{util}(\{x|Q'(x)\})}$$

That is to say that the probability of a user selecting a set corresponds to the total utility within the set relative to the total utility of all sets under consideration.

3.3 Decision procedure

Given a non-empty $Q(x)$, we generate a set of alternative clusterings, picking the one with *highest benefit*. Benefit is determined by the dividing information gain by the *cost* of summarizing the clusters to the user. Formally we pick $\hat{s}$ in:

$$\hat{s} = \arg\max_{s \in S}\left(\frac{\mathsf{gain}(\langle Q(x) : Q_1(x), ..., Q_m(x)\rangle)}{\mathsf{cost}(s)}\right)$$

where $s = \langle Q(x) : Q_1(x), .., Q_m(x)\rangle$ and S is the set of clustering statements. To keep things simple we assume that the cost of reporting the clustering $\langle Q(x) : Q_1(x), .., Q_m(x)\rangle$ is simply m. This assumption is of course too simplistic – a more reasonable measure, though beyond the scope of this paper, would be based on the cost of presenting the clustering in natural language.

Because S is (practically) infinite, we must give up on optimality and instead generate a representative sample $S' \subset S$. This set of clusterings is built randomly through splitting $Q(x)$ via new conditions and then by specializing (or generalizing) or further splitting the resulting clusters. Our methods to calculate gain are purely distributional. That is we directly compute our measures through iterating over answers sets yielded by our clusters. The calculation of gain is $O(m{\cdot}n^3)$ for the n options under consideration and a clustering of m clusters. Thus if we recast our problem as a search problem, the evaluation function is polynomial in the size of the problem. Although our current method to obtaining S' is still rather naive, such methods can achieve reasonable performance for moderately sized data sets.

[1]There are several other reasonable models that can be used here. For example the average utility or even a more complex measure of perceived utility based on discernibility.

Thus far we have left the set of conditions $\mathcal{C}$ unspecified. The conditions are just boolean mappings over options (or k-tuples). While the condition $\mathsf{PricelsPrime}(x)$ may return true for all options where the fifth component is a prime number, there are an infinite number of such far fetched conditions and thus we isolate attention to a fixed finite set of conditions $\mathcal{C}_{\mathsf{simple}} \subseteq \mathcal{C}$ which are the conditions that 'make sense' in the given domain.

Given $\mathcal{C}_{\mathsf{simple}}$, the set $\mathcal{Q}$ of semantically distinct queries that may be built up as boolean formulas from conditions within $\mathcal{C}_{\mathsf{simple}}$. Note that $\mathcal{Q}$ is large, though finite. We assume here that the natural language interface may relate user typed (or spoken) strings to elements within $\mathcal{Q}$ for the purposes of understanding and paraphrasing.

4 Example

Although we have a working demonstration system, we choose here to present a series of examples to illustrate the properties of our algorithm over the database of table 1.

4.1 Three user models and a model of discernibility

To simplify the presentation we assume a very simple linear user model based on the coefficients a_i and b_i. To achieve this we capture a value mapping function v_i for the i-th option component values to numerical measures: $v_i(\mathcal{U}) \rightarrow \mathbb{R}$. Assume that $v_i(z) = z$ for numerical values and $v_i(z) = 1$ for non-numerical values. The default utility of an option is thus measured as:

$$\mathsf{util}(t) = \sum_{i=1}^{k} a_i \cdot v_i(t[i]) + b_i$$

We present three user models. The first is for Maxwell Entropy: $a_i = 0, b_i = \frac{1}{7}$. As we can see, Max has no default preference for one option over another. The second user model is that of the student who only favors one option over another based on price: $a_i = 0, b_i = 0$ for $i \neq 5$, $a_5 = -1, b_5 = 300$. The third user model is that of a business traveler that prefers early flights and flights on SAS. As we shall see later this model is able to induce a tradeoff between options (e.g. early non-SAS

flights vs. later SAS flights): $a_i = 0, b_i = 0$ for $i < 3, i > 4$, $v_3('\mathsf{SAS}') = 1$, $v_3('\mathsf{Lufthansa}') = 0.1$, $v_3('\mathsf{AirFrance}') = 0.1$, $v_3('\mathsf{BritishAir}') = 0.1$, $a_3 = 1$, $b_3 = 0$, $a_4 = -.2$, $b_4 = 2.6$.

We assume the same model of discernibility for all users. For flight number and aircraft type we assume no ability of the users to discern between options, that is $\zeta_1(v_1, v_2) = 1$ and $\zeta_7(v_1, v_2) = 1$ for all value pairs v_1 and v_2. For destination we assume perfect discernibility, that is $\zeta_2(v_1, v_2) = 1$ when $v_1 = v_2$ and 0 otherwise. For airline and meal we assume strong discernibility, specifically $\zeta_3(v_1, v_2) = 1$ and $\zeta_6(v_1, v_2) = 1$ when $v_1 = v_2$ and .33 otherwise. For departure time and price we use an exponential measure. That is $\zeta_4(v_1, v_2) = e^{-|v_1 - v_2|}$ and $\zeta_5(v_1, v_2) = e^{-\frac{|v_1 - v_2|}{100}}$. Given this model, $\mathsf{sim}(t_1, t_4) = .074$.

4.2 System runs

Give the database of table 1 and the user and discernibility models above, table 2 shows the calculation of gain for various clusterings of the input query "the flights to Berlin leaving before noon." The highest benefit clusterings are presented along with several lower scoring alternatives to illustrate the sensitivity to the given user model. The key parameter that controls the behavior of the system is the penalty parameter γ.

5 Discussion

The work presented here is preliminary, likely to undergo much revision and refinement as it is integrated and evaluated within an operational NLI system [2]. Among the unsettled issues are the form and scope of the models of utility and discernibility. For example the independence assumption made in the current model of discernibility is likely to be inadequate in general. As an anonymous reviewer points out, a possible reason why SAS and Lufthansa are discernible could be that SAS provides meals while Lufthansa does not. We agree, although we note that a more sophisticated model could be developed and plugged into our basic approach. As for scope, we have assumed that preference and discernibility models can be crafted for large classes of users in a given context. For example we assume that under a wide variety of conditions flights on A320's and flights on A300's are indiscernible. Likewise we model flights to different locations as nearly perfectly discernible.

While perhaps our models of preference and discernibility should be generalized, we feel justified in stipulating the ideal answer assumption and our method of calculating information gain, confounded by discernibility and preference. A natural question is whether the ideal answer assumption can be relaxed. Our hypothesis is that the discernibility model and γ, the penalty of picking a non-ideal option, already provide enough machinery to get desired results. In any case, some variant of the ideal answer assumption seems necessary to cast the problem in an information theoretic light.

In contrast to *summarize-and-refine* based approaches [4], *user-modeling* based approaches, as characterized by [3, 6] rank matching options based on their utility, offering the highest ranking options first. Recently these two strategies (*summarize-and-refine* and *user-modeling*) have been combined into a single approach [1] based on building an *option tree* over the (current) set of options which specifies refinement paths based on a user model of attribute importance and attribute value preferences. Such option trees are further pruned based on dominance relations amongst options (i.e. when one option will always be preferred over another) and option trees are able to express trade-offs among options when the user has conflicting preferences (e.g. if the user prefers early flights and flights on SAS, then a response might be "At 8 am, flight 2 is the earliest flight to Berlin, but it's with Lufthansa, while flight 7, leaving at 10am, is the earliest flight to Berlin on SAS.") An elusive goal of the work in this paper, not yet achieved, is to provide an information theoretic account of why presenting such trade-offs yields especially high information gain.

6 Conclusions

We live in a time of tremendous choice; we pick from hundreds of mobile phone models, thousands of travel destinations and millions of potential chat partners. When confronted with such complex choices, people tend to become either *maximizers*, spending large amounts of time studying the various options, their features and trade-offs, or

user	response	penalty (γ)	benefit
M. Entropy	"The 2 flights with Lufthansa or 2 flights with SAS?"	3 bits	.39
	"Flight #7 (SAS at 10) or flight #8 (SAS at 11) or flight #2 (Lufthansa at 8)"	3 bits	.25
Student	"Flight #8 the cheapest or the 3 other more expensive flights."	3 bits	.52
	"Flight #8 the cheapest."	.1 bits	1.94
	"Flight #8 the cheapest or flight #5."	.1 bits	.89
Business	"Flight #2 (the earliest), Flight #7 (the earliest on SAS), or the remaining 2 flights?"	3 bits	.51

Table 2: Response to a request for "the flights to Berlin leaving before noon."

they become *satisfiers*, making snap decisions, often bad, but saving time and mental energy. This paper serves both these types through increasing the efficiency of finding high quality options. This paper has presented a method to uniformly measure clusterings that either generalizes the user's query or specializes the user's query or in fact some combination of such strategies. The higher the penalty parameter γ, the more the system will opt toward a maximizer strategy.

This paper follows in the tradition of *cooperative query answering* which seeks to provide the user with more natural answers. This paper has mainly developed a set of theoretical tools and have verified the reasonableness of the developed tool through a simple, distributional implementation of the said concepts. The work in this paper tends more toward *summarize-and-refine* methods than user modeling based techniques. One aspect that the system does not explore are the subtle issues of contrast and linguistic nuance in presenting results. The system follows the *summarize-and-refine* approach in this respect, providing relatively straight forward summarizations of the best clustering that are found. Future work aims toward building a more efficient algorithm to search for possible clusters and incorporating the work into a query paraphrasing and natural language interface system for end-user evaluation [2].

7 Acknowledgements

We thank Johanna Moore and Joseph Palifroni for a fruitful discussion of an earlier version of this work and Jutta Langel and Jurg Kohlas for reminding us of the power and elegance of information theory applied to databases, or, in their case, Information Algebras.

References

[1] V. Demberg and J. Moore. Information presentation in spoken dialogue systems. In *Proc of EACL*, pages 65–72, Trento, Italy, April 2006.

[2] M. Minock. A STEP towards realizing Codd's vision of rendezvous with the casual user. In *33rd International Conference on Very Large Data Bases (VLDB)*, Vienna, Austria, 2007. Demonstration session.

[3] J. Moore, M. Foster, O. Lemon, and M. White. Generating tailored, comparative descriptions in spoken dialogue. In *Proc. of the Seventeenth International Florida Artificial Intelligence Research Sociey Conference*. AAAI press, 2004.

[4] J. Polifroni, G. Chung, and S. Seneff. Towards automatic generation of mixed-initiative dialogue systems from web content. In *Proc. of Eurospeech '03*, pages 193–196, 2003.

[5] M. Walker, R. Passonneau, and J. Boland. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *Meeting of the Association for Computational Linguistics*, pages 515–522, 2001.

[6] M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore M., M. Johnston, and G. Vasireddy. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, 28:811–840, 2004.

Embodied anticipation in neurocomputational cognitive architectures for robotic agents

Alberto Montebelli, Robert Lowe, Tom Ziemke
(alberto.montebelli@his.se, robert.lowe@his.se, tom.ziemke@his.se)
University of Skövde, School of Humanities and Informatics
SE-541 28 Skövde, Sweden

Abstract

The coupling between a body (in an extended sense that encompasses both neural and non-neural dynamics) and its environment is here conceived as a critical substrate for cognition. We propose and discuss the plan for a neurocomputational cognitive architecture for robotic agents, so far implemented in its minimal form for supporting the behavior of a simple simulated robotic agent. A non-neural internal bodily mechanism (crucially characterized by a time scale much slower than the normal sensory-motor interactions of the robot with its environment) extends the cognitive potential of a system composed of purely reactive parts with a dynamic action selection mechanism and the capacity to integrate information over time. The same non-neural mechanism is the foundation for a novel, minimalist anticipatory architecture, implementing our *bodily-anticipation hypothesis* and capable of swift re-adaptation to related yet novel tasks.[1]

Keywords: cognitive robotics; embodied cognition; dynamic systems; neuromodulation; anticipation; multiple time scales; bio-regulation.

[1]This work is a revised recombination of the following papers:

1. A. Montebelli, R. Lowe and T. Ziemke. The cognitive body: from dynamic modulation to anticipation. In G. Pezzulo, M. Butz, O. Sigaud, and G. Baldassarre, editors, Anticipatory Behavior in Adaptive Learning Systems. Berlin, 2009. Springer (in press).

2. A. Montebelli, R. Lowe and T. Ziemke. Embodied anticipation for swift re-adaptation in neurocomputational cognitive architectures for robotic agents. CogSci 2009 (in press).

1 Towards a cognitive robotic rendition of emotions

A systemic approach to the study of cognition permeates the seminal work of early cybernetics (Ashby, 1952; Wiener, 1965). In its modern form, the idea that *the whole is more than (and qualitatively different from) the sum of its parts* received a sound mathematical formalization through the science of non-linear dynamic systems (e.g., Bergé et al., 1984; Haken, 2004) and pragmatic validation through physics. It constitutes one of the core theoretical milestones of contemporary science and influenced cognitive science with a whole new scientific paradigm, namely the Dynamic Systems approach to the study of biological cognition (e.g., see Van Gelder, 2000; Kelso, 1995; Thelen & Smith, 1996).

The critical revision of the roles of body and environment in the cognitive process (e.g., Froese & Ziemke, 2009) constitutes the fundamental idea behind our paper. The systemic view conceives body and environment of the cognitive agent as constitutive of a largely distributed cognitive process, backing the brain in its operation by constantly offering cognitive support and tools (Clark, 2008). Thus, the cognitive process is the result of the activity of the brain-body-environment triad, whose components, coupled in a global dynamic, are equally necessary to the creation of the mental process (Kelso, 1995; Clark, 1997). The body can be interpreted as an enduring pre/post-processor of neural information (Chiel & Beer, 1997), and its interaction with the environment stores a wealth of knowledge about the "how to" of a cognitive activity (Pfeifer & Bongard, 2007). Research in embodied and situated cognition investigates in theoretical and experimental terms the role of the body and of the environment in the cognitive pro-

cess (Varela et al., 1992; Ziemke et al., 2007; Clancey, 1997). In this light cognitive robotics, i.e., the use of robots as models of embodied and situated cognition, is the perfect candidate for generating an experimentally grounded synthesis, as it forces us researchers to take very seriously the interplay among coupled bodies, control systems and environments (Parisi, 2004; Ziemke & Lowe, 2009).

Alongside the role of the body projected towards its environment, there is a less obvious, less visible and consequently often neglected internal dynamic component of the body. We are referring to the plethora of background bio-regulatory mechanisms, aimed at the maintenance of a viable metabolic balance necessary for the organism's survival. An increasing number of researchers investigate the potential cognitive role of this hidden dynamic. Antonio Damasio illustrates a view of cognition deeply rooted in a hierarchy of bodily processes and consistent with state-of-the-art neurological findings (Damasio, 2000, 2003). According to Damasio, emotions emerge from the complex hierarchy that constitutes the levels of *automated homeostatic regulation* - the basic evolutionarily determined organization for the maintenance of the living organism (ref. Figure 1). Metabolic regulation (e.g., endocrine/hormonal secretion, muscle contraction facilitating digestion), basic reflexes (e.g., basic tropism or taxes) and the immune system constitute the lower level of the machine. At a higher hierarchical level come behaviors related to pleasure/reward or pain/punishment (e.g. feeling pain triggers a specific pattern of protective behaviors), drives and motivations (e.g., hunger, thirst, curiosity, play and sex). One step further in the hierarchy we find *emotion proper* (e.g., joy, sorrow, fear) as a subset of the homeostatic reactions that is triggered by *emotionally competent stimuli* (ECS), either actual or imagined. ECS are such in virtue of the evolutionary history or of the ontogenesis of the organism. Finally, at the top of the hierarchy, from the current body state mapped in cortical body maps emerge (either conscious or unconscious) *feelings*. Feelings are perceptions of a certain state of the body, together with the perception of a certain mode of thinking and of attuned thoughts with certain themes. Similar approaches constitute the core motivations of *somatic theories of emotions* (Prinz, 2004; Panksepp, 2005).

Indeed, grounding emotions in physical (rather than mental) terms constitutes a possible entry point for their appealing robotic rendition. In a recent paper, Domenico Parisi points to the necessity of a deep investigation of the relation between the control system and what happens inside of the body (Parisi, 2004). The emphasis on bodily parameters affecting bodily processes can be traced back further to the cyberneticist W. Ross Ashby, who focused on the behavioral consequences of a set of *essential variables*, critical to the organism's survival (e.g. sugar concentration in the blood and body temperature). According to Ashby, the organism's need to restrict their range within viable limits determines the onset of a random creation of new adaptive behaviors (Ashby, 1952). Focusing on the cognitive implications of bio-regulatory processes might be a promising direction for scientific explorations in order to implement robots endowed with genuine autonomy, agency, intentionality and meaningful interaction with their environment (Ziemke & Lowe, 2009; Ziemke, 2008; Lowe et al., 2008). Indeed, internal robotics in the *here and now* is not sufficient for modeling emotions. It requires the presence of emotionally competent stimuli that derive from the coupling of body and environment in an *adaptive history of interactions*. This interpretation of internal robotics informs the particular approach described in this paper.

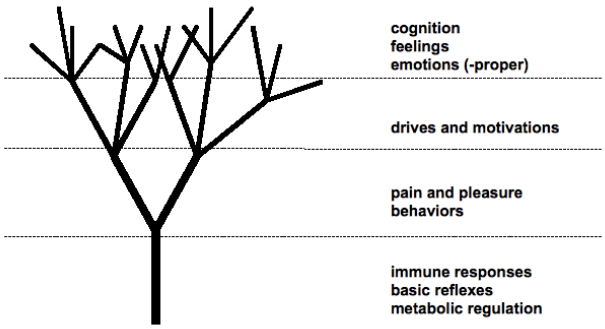


Figure 1: Damasio's representation of the levels of automated homeostatic regulation. Adapted from (Damasio, 2003).

As a matter of fact, all the above is in contrast to the traditional perspective on AI and cognitive science, i.e., the presumption that the description of the world in terms of related symbol structures and logical processing on such structures is the necessary and sufficient condition for general intelligent action by appropriate instances of physical systems (Newell, 1980). A concept mapped in cognitive robotics onto the linear *sense-plan-execute* scheme, and conceptually akin

to the functional approach of traditional computational neuroscience, focused on specific and decontextualized subdomains.

2 From bodily neuromodulation to bodily anticipation

In recent minimalist cognitive robotics experiments we tested two different experimental scenarios (for detail, see Montebelli et al., 2008, 2007, 2009). In both experiments a simulated Khepera robot was free to move in a square arena, where two identical light sources, centrally located in the environment, cast a stationary light gradient. An invisible recharging area was centered under one of the two lights, randomly selected for each replication. The robot received sensory information through its light and distance sensors and moved according to the activation of two wheels controlled by a simple sensory-motor map, i.e., a single-layer, feed-forward artificial neural network (ANN). It also sensed its simulated energy level (e.g., the level of a battery charge), subject to linear decay, from a maximum value down to zero. In both scenarios, the fitness function rewarded at each time step the maintenance of positive levels of energy. Each individual was tested on runs of constant duration, for several replications. At the end of each generation, the best individuals were selected for reproduction according to a standard evolutionary algorithm.

2.1 Experiment 1

The entering of the recharging area area provided an instantaneous full energy recharge. The evolutionary algorithm evolved weights and biases of the ANN.

Obviously, the evolved agents performed well on such an elementary task. The interesting part of our work came when, setting aside the evolutionary task, we selected the best individual and used its energy level as control parameter of the agent-environment system. We clamped the energy level to a fixed value for the whole duration of each replication, and systematically explored values from empty to full in the different replications. Consequently, we were able to map the behavioral repertoire of the evolved agent as a function of its energy level. We observed three main classes of behavioral attractors (ref. Figure 2, left): *exploratory behaviors* (i.e., the agent engages in large loops from one light source to the other - attractor class 'A'), *local behaviors* (the agent's loops are closely bound to a single light source - class 'C') and *hybrid behaviors* (combining the characteristics of both exploratory and local attractors - class 'B'). The expression of these three behavioral attractors was neatly distributed as a function of the energy level (ref. Figure 2, right). Exploratory behaviors dominated the lowest range of energy levels, whereas local behaviors the highest ones. For intermediate levels of energy we found the prevalence of hybrid behaviors.

In sum, we showed how: 1) Minimalist non-neural bodily states (e.g., the energy level in our experiment) can *modulate* the sensory-motor map implemented by an ANN, and thus the behavior of the simulated robotic agent coupled with its environment. 2) This modulation can be exploited as a *dynamic action selection mechanism*. During the evolutionary task different classes of behavioral attractors were locally available to the agent, depending on its energy level. For example, an energy level of 0.7 (ref. Figure 2, right), led to the expression of attractor C3 (in 70% of the replications), C1 (20%) or B1 (10%). The actual selection of the specific attractor depended on the basin of attraction in which the combination of the starting position and the integrated effects of noise induced the system dynamics. 3) The cooperation between dynamics at different *time scales* can boost the cognitive potential of the system. In the case of our experiment (where the energy level mechanism was one order of magnitude slower than the normal sensory-motor interactions), a collection of purely reactive components was endowed with the capacity to integrate information over time (see Discussion).

2.2 Experiment 2

As before, a stationary gradient of environmental luminance (continuous sensory regime), correlated with a rewarding area centered on a randomly selected light source. However, during each replication this regime alternated with an intermittent sensory regime, where the light sources were obscured every third time step. Under this new condition, the randomly chosen area determined a punishment in the form of an energy leak. As a biological metaphor, this alternation between regimes models the case of a succulent berry whose external pigmentation is different when unripe (and toxic) or ripe (and energizing). Again, the goal consisted in maintaining a positive energy level.

We compared the simple architecture described in the

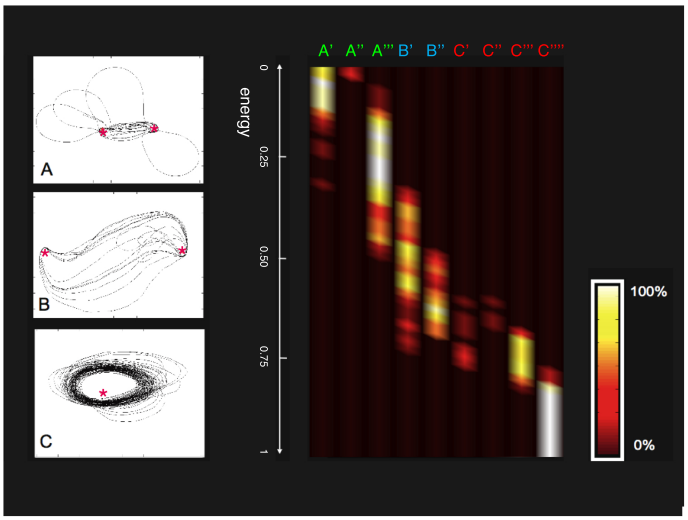


Figure 2: **left-** Sample spatial trajectories for the three classes of behaviors observed in clamped conditions after transient exhaustion. Exploratory behaviors (panel A), local behaviors (panel C) and hybrid forms (panel B). The position of the light sources are indicated by red stars. **right-** The intensity of the pixels for each column (corresponding to attractors belonging to classes A-C, as specified by their labels on the top row) represents the relative frequency of the behavioral attractor as a function of the energy level. For energy levels in the interval [0.0, 0.4] we can observe a clear dominance of attractors in class A. Attractors in class C dominante in the energy interval [0.7, 1.0]. Data from 500 replications (10 for each energy level). Adapted from (Montebelli et al., 2008).

previous experiment with a novel minimalist anticipatory architecture. In the former case, the evolutionary algorithm adapted the ANN's weights and biases on the new task, starting either from the final population evolved in the previous experiment or from a randomly generated population. In the case of the new architecture, shown in Figure 3, the original ANN (i.e., the simple ANN, whose weights and biases were extracted adopting the final population evolved during the previous experiment) was backed by a pre-adapted *mixture of recurrent experts* (Tani & Nolfi, 1999) that processed the sensory flow. During its adaptation, each expert competed with the others in order to generate the best prediction of the sensory state at the next time step. By doing so, two different experts became specialized by tuning to the specific dynamic flow of the two different regimes. Crucially, in the new architecture the activation of the expert tuned to the intermittent sensory regime triggered a new energy mechanism that overrode the original one. The decay rate of the overriding energy mechanism, rather than hardwired as before, is the one single parameter adapted by an evolutionary algorithm on the new task.

In short, we found that: 1) The systems provided with the anticipatory architecture developed an effective dynamic relation with its environment. They demonstrated a straightforward engagement with the rewarding light source during the continuous sensory regime, and a swift disengagement from the penalizing one during intermittent regime (ref. Figure 4, bottom). On the other hand, systems provided with the original ANN architectures tended to cope with the new task by relying on stereotypical behavioral attractors (Figure 4, top). During the continuous sensory regime they engaged in loops containing both light sources, approaching them close enough to enter their potential rewarding areas. During the intermittent regime they simply relaxed their trajectories with respect to the light sources, keeping at a slightly larger distance from them and consequently clear from the critical area, thus avoiding the punishment. This behavior ignores the effect of the recharging area on the energy level, merely relying on light sensor information and geometrical constraints. 2) In the case of the anticipatory architecture, the adaptive process for the new task proved easy, as even a random search could immediately generate agents with satisfac-

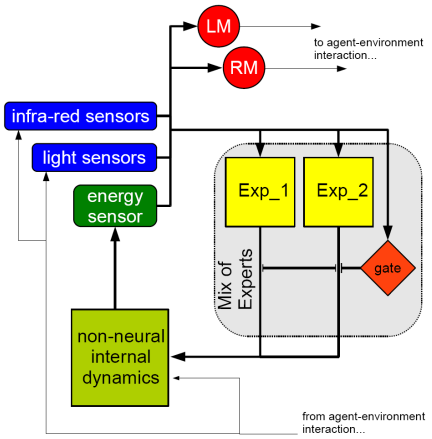


Figure 3: Minimalist anticipatory architecture. The sensory information (infra-red, light and energy sensors) drives the left and right motors (LM and RM) through a feedforward ANN with no hidden layers. The sensory flow is also processed by a mixture of recurrent experts, pre-adapted so that each expert is tuned to a specific sensory regime. The information on the current best expert (corresponding to one of the two regimes) is given by the gating signal, that selects the current energy mechanism of the agent. Adapted from (Montebelli et al., 2009).

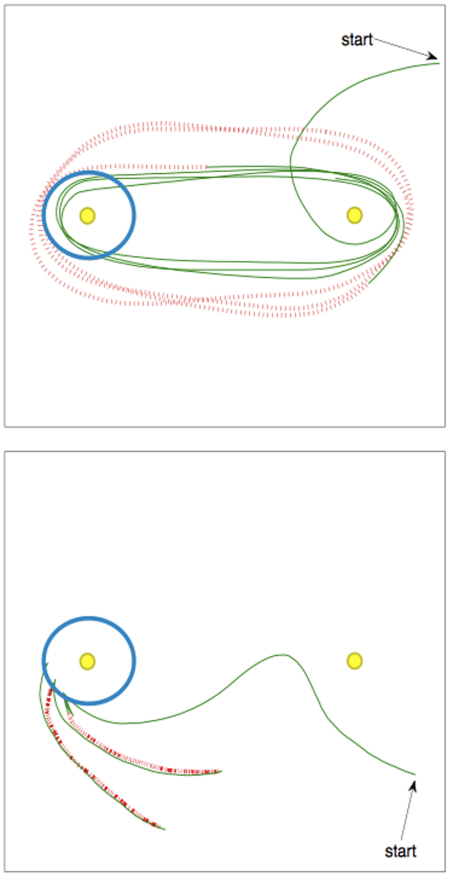


Figure 4: Prototypical spatial trajectories developed by the different architectures during evolutionary adaptation. **top -** Agents provided with simple feedforward ANNs tended to deploy a stereotypical strategy, i.e., their trajectories systematically engaged in exploratory loops between the two light sources, entering the recharging area (leftmost circle) during the continuous regime (continuous line) and avoiding it during the intermittent regime (dashed line). **bottom -** On the other hand, our anticipatory architecture showed dynamical engagement and disengagement with the rewarding/punishing area according to the different sensory regime (again, continuous/dashed lines represent the trajectories during continuous/intermittent sensory regimes). Adapted from (Montebelli et al., 2009).

tory performance. The evolutionary search was much more problematic for the original ANN, evolved from both starting conditions.

2.3 An initial synthesis: the bodily-anticipation hypothesis

We will try to formalize the previous results in a general scheme. We have just seen how non-neural internal dynamics can modulate the current modality of the agent-environment interaction (i.e., its current behavioral attractor). On the other hand, the current behavior determines the current non-neural internal dynamics (e.g., an effective behavior that satisfies the experimental task maintains a high energy level). This bidirectional relation is expressed by the arrows connecting the blocks labeled SENSORY-MOTOR FLOW and NON-NEURAL INTERNAL DYNAMICS in Figure 5. The former block represents the dynamic of the degrees of freedom relevant to the current sensory-motor engagement between the agent and its environment. Similarly, the latter embeds the relevant non-neural internal dynamics. In parallel, current sensory motor flow and internal

dynamic drive a neural emulator block (labeled ANTICIPATION) that is capable, in virtue of its evolutionary history and/or ontogenetic adaptation, of dynamic anticipation. We suggested elsewhere (Montebelli et al., 2009) that a cognitive system settled on its behavioral attractor constitutes an important instance of an implicitly anticipatory system. In fact, the engagement with the attractor binds the system to a stable and qualitatively determined dynamic flow. An autonomous and viable dynamic is inherently endowed with anticipatory power. The main practical function of this emulator is to tune to the current sensory-motor dynamic and dynamically perturb the bodily dynamics with the anticipated consequences of the current dynamic interaction.

For example, consider a specimen agent, a caveman engaged in a relaxing and innocuous activity, e.g., picking berries in a forest. Out of the blue, an emotional stimulus, e.g., an apparently hungry, massive dinosaur, loudly enters the scene. The enormous time gap that separates the extinction of dinosaurs and the appearance of the first hominids is part of our example. We want to make sure that our specimen is experiencing a novel situation (therefore, a positivist caveman, who only brings solid scientific arguments to prove the dinosaur's anachronism, would be the perfect candidate for premature exhaustion of his own pedagogical role). The caveman's anticipatory system has no difficulty in predicting the most likely future scenario. The sensory-motor flow correspondent to the ongoing activity (picking berries) must be inhibited and redirected to a more conservative attitude. How will the next viable behavior (e.g., an impulsive fleeing) be selected? With this question in mind, our experiment explored the feasibility of a body-mediated pathway (arrow a-b in Figure5). We tested the hypothesis that the anticipatory block (minimally implemented as the mixture of recurrent experts) might directly influence the non-neural bodily dynamics. In our prehistoric example, that means that once he perceived the emotional stimulus, our caveman would physically experience his own body torn by the fangs and nails of the dinosaur. It is likely that the caveman's evolutionary history and his ontogenesis had already created viable correlations between his dramatic visceral reaction and his fleeing for life, although the specific situation had never been experienced before. This constitutes the essence of our *bodily-anticipation hypothesis*: the selection of the next viable action is offloaded onto the bio-regulatory dynamics of the body. Destabilized by the anticipated effect of the current interaction, the body reacts *as if* actually engaged in such sensory-motor experience. The bodily perturbation elicits reactions, already stored in the potential of bodily and neural interactions, that tend to pull the system back into viable regions.

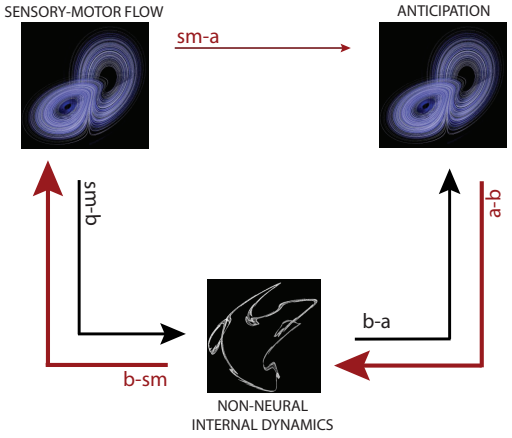


Figure 5: Illustration of the bodily-anticipation hypothesis. During its daily roaming, our agent gets engaged with a potentially noxious interaction. Neural sensory-motor anticipatory dynamics, here conveniently isolated within the global coupled system (box labeled ANTICIPATION), predict the risk by physically perturbing the current non-neural bodily dynamics (NON-NEURAL INTERNAL DYNAMICS) through path a-b and from there, indirectly through a further path b-sm, the actual sensory-motor dynamics (SENSORY-MOTOR FLOW). Following a quick reorganization of its behavioral attractor, our agent is attuned to face the novel danger thanks to the mediation of its body, without any direct influence of anticipation on the selection of the new behavior. Adapted from (Montebelli et al., 2009).

3 Discussion

3.1 On the internal/external dichotomy

We hope to have clarified enough the importance of conceptualizing the phenomenon of cognition as emergent from the coupling of body (with its external morphology and the richness of its internal bio-regulatory mechanisms), nervous system and environment. Within this systemic view, the boundary separating each subsystem is nothing but a useful artifice, functional to the

analysis of a complex system dominated by circular relations. Each component participates in the global cognitive process with equal weight. In this sense, even defending the traditional labels of cognitive robotics, where the nervous system would be assimilated the control system, would be problematic. What is controlled? What is doing the controlling? From our example it seems clear enough that different parts of the system mutually influence and are influenced by others (e.g., the energy level can modulate the behavior of the sensory-motor map, that in return affects the energy level).

This tight coupling casts a light an interesting point. What is internal? What is external? Of course we have no difficulty at drawing a line from our distal, anthropomorphic perspective. Nevertheless we can easily argue that a simple agent, even substantially more complex than our elementary model, might find defining such a boundary difficult. We prefer to avoid such dichotomy, as we consider more useful focusing on the global system composed of dynamically interacting parts. At any given time its dynamic balance will be perturbed by stimuli coming from different sources (e.g. the external environment, the agent's regulatory mechanisms, its nervous system). Each perturbation would produce a consonant reaction of the system's trajectory in its phase space. Each time, according to the needs of the analysis, we will have to properly redraw the boundary between input and output, cause and its effect. Parisi suggested objective criteria for partitioning the inside and outside of the body in natural agents, on the grounds of the physical-chemical processes that tend to dominate the two interfaces (Parisi, 2004). Local and specific interactions with fast dynamics, archetypal of physical processes, tend to characterize the interface with the external world. Global and diffused variations with slower time scales, characteristic of chemical processes, tend to take place inside the organisms. Although this is just a generalization, the focus on the different time scales prepares us for the next fundamental observation.

3.2 On the role of multiple time-scales

An obvious objection can be raised against our model. What is it that determines the distinction between neural and non-neural? Could the non-neural internal dynamic be translated into purely neural mechanisms? After all, the work of other groups (e.g., Tani & Ito, 2003; Ito et al., 2006; Tani & Nolfi, 1999) seems oriented in that direction.

Rather than taking a defensive stance, we will simply redirect the problem and dissolve it in its abstract formalization. The interplay of the different time scales that characterize the energy mechanism and the other sensory-motor interactions with the environment is crucial to our model. In the experiment reported in Section 2.1, during the artificial evolution of the system, the slower dynamic of the energy level organized the continuous sensory-motor flow in dynamically related events. This endowed the system, composed of purely reactive elements, with the capacity to integrate information over time. Elsewhere (Montebelli et al., 2009), we conjectured that: "...The access to a collection of attuned dynamic sub-systems characterized by intrinsic dynamics at different time scales and the exploitation of such differences, constitutes a powerful mechanism of embodied cognition, widely operating at the different levels of organization of biological cognition. A mechanism providing the cognitive system with the capacity to structure information on events which are relevant to its survival, with no need for explicit representations, memory or consciousness." With this in mind we can look at the plethora of bio-regulatory phenomena with new eyes. The characteristic time scales of non-neural bodily processes, so different from the normal dynamics of the sensory-motor interactions between an agent and its environment, might provide exactly that dynamical richness that we are advocating. The role of multiple time scales is currently attracting the attention of the scientific community, both in computational neuroscience (e.g., Kiebel et al., 2008; Fusi et al., 2007) and cognitive robotics (e.g. Yamashita & Tani, 2008; Ito et al., 2006; Paine & Tani, 2005; Tani & Nolfi, 1999).

3.3 Experimental evidence for the bodily-anticipation hypothesis

The paths in the general scheme sketched in Figure 5 are actually less arbitrary than they might look at first glance. In the present subsection, we report some experimental evidence that supports our bodily-anticipation hypothesis, from natural and artificial systems. Our own and related work in cognitive robotics (Montebelli et al., 2008, 2007; Tani & Ito, 2003; Ito et al., 2006), motivates the arrows representing the relation between the non-neural internal dynamics and the sensory-motor flow blocks (paths sm-b and b-sm). The claim that in organisms the internal dynamics of the body (e.g., a sud-

den injection of adrenaline) affect the behavior and that behavior affects the body (e.g., eating or declining the fifth slice of your birthday cake) shouldn't strike us as bizarre. The capacity of the brain to anticipate sensory-motor correlates (path sm-a) is currently the object of intensive research in neuroscience (e.g., see Hesslow, 2002). Examples in cognitive robotics are in (Tani & Nolfi, 1999; Ito et al., 2006). Interestingly, Ziemke et al. show how a viable anticipation does not have to be identical to the anticipated phenomenon (Ziemke et al., 2005). An example of how a neural event taking place in the nervous system, might affect the body is given in (Damasio, 2000): the case of a professional musician is reported, who could systematically control her emotional machinery in experimental conditions. Also the seemingly arbitrary switch between the natural energy dynamic and the overriding energy mechanism taking over during the intermittent sensory regime is inspired by neurophysiological analogs. False bodily information can sometimes substitute for the actual state, for example, in the case of endogenously altered nociceptive signals. There is an obvious advantage for a wounded organism to ignoring the pain when it is fleeing from the danger that produced it (Damasio, 2003).

3.4 The body for search-space compression

Obviously, our bodily-anticipation hypothesis does not rule out the possibility of a co-existence with a neural pathway between anticipation and sensory-motor flow (the missing path a-sm in Figure 5). Nevertheless, we point to the fact that our minimalist anticipatory architecture drastically simplifies the problem of readapting to a new task. Our proposal focuses on the knowledge that is already embedded in the body after the long history of biological evolution and ontogenesis, and might be exploited during readaptation. The search space during readaptation, characterized by the potentially enormous number of degrees of freedom of an ANN, is reduced by our bodily-anticipation hypothesis to the much smaller dimensionality of the bodily neuromodulators (the energy level in our minimalist example). We believe that the bodily-anticipation hypothesis could be of help at least in virtue of such drastic compression of the adaptive search space, particulary in circumstances that require, for example, fast, non-deliberated decision making. Rather than searching the massive space of the system's degree of freedom for the proper associations supporting the a-sm pathway, the system can limit its exploration to the subspace of the bodily parameters. Pragmatically, even a random search of the appropriate decay rate of the overriding energy dynamic in our anticipatory architecture can swiftly readapt the system to the new problem, whereas such readaptation proves slow with the original architecture. This is obviously related to Ashby's work on *ultrastable agents*. A random change in the behavioral coupling between the agent and its environment is induced whenever a variation of an *essential variable* threatens its survival (Ashby, 1952; Di Paolo, 2003).

An argument in favor of a mental path seems to be brought forth by Damasio, as he introduces the *as-if body loops* (Damasio, 2000). The emotional machine, grounded in the homeostatic process as introduced in Section 1, is in Damasio's theory central even to highly logical functions, e.g. decision making (Damasio, 2000). Its support can be elicited directly, but after repeated exposure the brain can build consistent causal associations and thus totally bypass the body in the decision process. Nevertheless, Bechara refers to preliminary results suggesting how in the process of decision making the role of the as-if body loop might be restricted to the most predictable situations (choice under certainty). As the decision scenario drifts towards risk or ambiguity (full uncertainty), a mode of operation where the bodily mechanisms are directly engaged becomes prominent (Bechara, 2004). We find this observation perfectly tuned with the intuition inspiring our model.

3.5 Future work

We consider our minimal anticipatory architecture as a promising and complete illustration of our bodily-anticipation hypothesis, although still at its initial stage of development. Nevertheless, together with a few answers, it suggests plenty of supplementary questions. Accordingly, we admit that it needs and deserves further investigation and validation.

Our model might be accused of being an *ad hoc* arrangement, built on the basis of the previous experiment. In other words, it might be suspected that we embed built-in solutions in our minimalist anticipatory architecture: First, for the arbitrary decision to override the original non-neural internal mechanism (although we have demonstrated in the previous subsection how the same strategy can be found in natural agents); Second, for selecting the decay rate of the overriding en-

ergy mechanism as critical parameter to be adapted by the evolutionary algorithm. This is a reasonable criticism. Nevertheless, given the extreme simplicity of our current setup, such design choices were necessary. In our model, simplicity constitutes a deliberate preference. For the sake of a detailed analysis, we try to implement the minimal model capable of producing the phenomenon under study.

However, we welcome such objection, confident that it can be more easily confuted given a slightly more complex model, both in terms of task and architecture. In particular, future work will specifically address the implementation of more realistic internal dynamics, inspired by natural metabolic systems as well as by the work on prototypical robotic agents endowed with *microbial fuel cells* (Melhuish et al., 2006).

4 Conclusions

This paper takes on and extends the tradition of a more systemic view of AI research (e.g., Montebelli et al., 2008; Froese & Ziemke, 2009; Ziemke & Lowe, 2009). Cognition is conceived and analyzed in terms of coupled systems: the body (encompassing both its external morphology and its internal bio-regulatory mechanisms), the nervous system and the environment constitute a cognitive aggregate. Such interpretation dissolves the internal-external dichotomy into a formalization in terms of coordinated multiple time-scales. The cognitive role of the body is taken in account with special and novel emphasis on what happens inside of the body. Biological cognition, more than simply inspiring problems and solutions, is seen as the living implementation of the basic organizational principles of intelligence, still mostly to be unraveled.

In a first experiment (ref. Section 2.1) we showed how non-neural internal dynamics, following a slow time scale, can modulate the activity of an ANN and consequently the behavior of an agent coupled with its environment. A traditional evolutionary algorithm self-organized this modulation, implementing a dynamic action selection mechanism. The analysis showed how the coordination of multiple time-scales might support the emergence of more sophisticated cognitive capacities. In a second experiment (Section 2.2) we extended the previous system to a novel anticipatory architecture, providing a minimalist implementation of the bodily-anticipation hypothesis presented in this paper. The novel architecture provided flexible and dynamic engagement of the agent with its environment, as a swift re-adaptation to a brand new task was accomplished. Crucially, the search for novel behaviors was drastically simplified, as it operated on the limited subspace of the non-neural internal parameters, rather than on the high dimensional space of the ANN. We believe that this work illustrates promising results in terms of basic organizational principles of cognition that can be usefully explored by minimally cognitive architectures.

Acknowledgments

This work has been supported by a European Commission grant to the project *Integrating Cognition, Emotion and Autonomy* (ICEA, www.iceaproject.eu IST-027819) as part of the *European Cognitive Systems* initiative.

References

Ashby, W. R. (1952). *Design for a brain: The origin of adaptive behavior*. London: Chapman '&' Hall.

Bechara, A. (2004). The role of emotion in decision-making. *Brain and Cognition*(55), 30-40.

Bergé, P., Pomeau, Y., & Vidal, C. (1984). *Order within chaos*. Wiley-Interscience.

Chiel, H., & Beer, R. D. (1997). The brain has a body. *Trends in Neurosciences*, *20*(12), 553-557.

Clancey, W. J. (1997). *Situated cognition*. Cambridge University Press.

Clark, A. (1997). *Being there: Putting brain, body, and world together again*. Cambridge, MA: MIT Press.

Clark, A. (2008). *Supersizing the mind*. Oxford University Press.

Damasio, A. (2000). *The feeling of what happens*. Harvest Books.

Damasio, A. (2003). *Looking for Spinoza*. Harcourt.

Di Paolo, E. (2003). Organismically-inspired robotics. In K. Murase & T. Asakura (Eds.), *Dynamical systems approach to embodiment and sociality* (p. 19-42). Adelaide: Advanced Knowledge International.

Froese, T., & Ziemke, T. (2009). Enactive artificial intelligence. *Artificial Intelligence*, *173*, 466-500.

Fusi, S., Asaad, W. F., Miller, E. K., & Wang, X.-J. (2007). A neural circuit model of flexible sensorimotor mapping. *Neuron*, *54*(2), 319-333.

Haken, H. (2004). *Synergetics: introduction and advanced topics*. Springer.

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences*, *6*(6), 242-247.

Ito, M., Noda, K., Hoshino, Y., & Tani, J. (2006). Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. *Neural Networks*, *19*(3), 323-337.

Kelso, J. A. S. (1995). *Dynamic patterns*. Cambridge, MA: MIT Press.

Kiebel, S. J., Daunizeau, J., & Friston, K. J. (2008). A hierarchy of time-scales and the brain. *PLoS Computational Biology*, *4*(11).

Lowe, R., Herrera, C., Morse, A., & Ziemke, T. (2008). The embodied dynamics of emotion, appraisal and attention. In L. Paletta & E. Rome (Eds.), *Attention in cognitive systems* (p. 1-20). Berlin: Springer.

Melhuish, C., Ieropoulos, I., Greenman, J., & Horsfield, I. (2006). Energetically autonomous robots: food for thought. *Autonomous Robots*, *21*, 187-198.

Montebelli, A., Herrera, C., & Ziemke, T. (2007). An analysis of behavioral attractor dynamics. In F. Almeida e Costa (Ed.), *Advances in artificial life: Proceedings of the 9th european conference on artificial life* (p. 213-222). Berlin: Springer.

Montebelli, A., Herrera, C., & Ziemke, T. (2008). On cognition as dynamical coupling: An analysis of behavioral attractor dynamics. *Adaptive Behavior*, *16*(2-3), 182-195.

Montebelli, A., Lowe, R., & Ziemke, T. (2009). The cognitive body: from dynamic modulation to anticipation. In *Anticipatory behavior in adaptive learning systems.* Springer- in press.

Newell, A. (1980). Physical symbol systems. *Cognitive Science*, *4*(2), 135-183.

Paine, R. W., & Tani, J. (2005). How hierarchical control self-organizes in artificial adaptive systems. *Adaptive Behavior*, *13*(3), 211-225.

Panksepp, J. (2005). Affective consciousness. *Consciousness and Cognition*, *14*, 30-80.

Parisi, D. (2004). Internal robotics. *Connection Science*, *16*(4), 325-338.

Pfeifer, R., & Bongard, J. (2007). *How the body shapes the way we think*. Cambridge, MA: MIT Press.

Prinz, J. J. (2004). *Gut reactions*. Oxford University Press.

Tani, J., & Ito, M. (2003). Self-organization of behavioral primitives as multiple attractor dynamics. *IEEE Trans. on Systems, Man, and Cybernetics. Part B*, *33*(4), 481-488.

Tani, J., & Nolfi, S. (1999). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks*, *12*(7-8), 1131-1141.

Thelen, E., & Smith, L. B. (1996). *A dynamic systems approach to the development of cognition and action*. MIT Press.

Van Gelder, T. (2000). The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences*, *21*, 615-628.

Varela, F. J., Thompson, E. T., & Rosch, E. (1992). *The embodied mind*. MIT Press.

Wiener, N. (1965). *Cybernetics, or control and communication in the animal and the machine*. MIT Press.

Yamashita, Y., & Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model. *PLoS Computational Biology*, *4*(11).

Ziemke, T. (2008). On the role of emotion in biological and robotic autonomy. *BioSystems*, *91*, 401-408.

Ziemke, T., Hesslow, G., & Jirenhed, D. A. (2005). Internal simulation of perception. *Neurocomputing*, *68*, 85-104.

Ziemke, T., & Lowe, R. (2009). On the role of emotion in embodied cognitive architectures: From organisms to robots. *Cognitive computation*, *1*(1), 104-117.

Ziemke, T., Zlatev, J., & Frank, R. M. (Eds.). (2007). *Body, language and mind: Embodiment* (Vol. 1). Berlin/New York: Mouton de Gruyter.

Integrating Case-Based Inference and Approximate Reasoning for Decision Making under Uncertainty

Ning Xiong, Peter Funk

School of Innovation, Design and Engineering
Mälardalen University
SE-72123 Västerås, Sweden
{Ning.Xiong, Peter.Funk}@mdh.se

Abstract. This paper proposes a novel approach to decision analysis with uncertainty based on integrated case-based inference and approximate reasoning. The strength of case-based inference is utilized for building a situation dependent decision model without complete domain knowledge. This is achieved by deriving states probabilities and general utility estimates from the subset of retrieved cases and the case library given a situation in query. In particular, the derivation of state probabilities is realized through an approximate reasoning process which comprises evidence (case) combination using the Dempster-Shafer theory and Bayesian probabilistic computation. The decision model learnt from previous cases is further exploited using decision theory to identify the most promising, secured, and rational choices. We have also studied the issue of imprecise representations of utility in individual cases and explained how fuzzy decision analysis can be conducted when case specific utilities are assigned with fuzzy data.

Keywords: decision analysis with uncertainty, case-based inference, decision model, basic probability assignment, approximate reasoning.

Planning Speech Acts in a Logic of Action and Change*

Martin Magnusson and Patrick Doherty
Department of Computer and Information Science
Linköping University, 581 83 Linköping, Sweden
{marma,patdo}@ida.liu.se

Abstract

Cooperation is a complex task that necessarily involves communication and reasoning about others' intentions and beliefs. Multi-agent communication languages aid designers of cooperating robots through standardized speech acts, sometimes including a formal semantics. But a more direct approach would be to have the robots plan both regular and communicative actions themselves. We show how two robots with heterogeneous capabilities can autonomously decide to cooperate when faced with a task that would otherwise be impossible. Request and inform speech acts are formulated in the same first-order logic of action and change as is used for regular actions. This is made possible by treating the contents of communicative actions as quoted formulas of the same language. The robot agents then use a natural deduction theorem prover to generate cooperative plans for an example scenario by reasoning directly with the axioms of the theory.

1 Introduction

Autonomous agents reason about the world to form plans and affect the world by executing those plans. Thus, agents' plans have an indirect effect on the world, and it becomes important for reasoning agents to take other agents' plans into account. Furthermore, they would do well to plan actions that affect other agent's plans and thereby (doubly indirectly) affect the world. Philosophers of linguistics have realized that we humans do this all the time through communication. In particular, Searle's *speech acts* [24] characterize natural language utterances as actions with conditions upon their execution and effects on the mental states of others.

Perrault, Allen, and Cohen [19] establish a useful connection between speech acts and planning. They formalize speech acts as planning operators in a multi-modal logic of belief and intention. Using these an agent can *inform* other agents about some fact, *request* other agents to perform some action, or ask other agents questions by requesting them to inform about some fact. They encoded simplified versions of these actions as STRIPS-like planning operators and used a backward-chaining algorithm to generate plans involving both regular actions and speech acts.

Research on software agents [8] has also adopted speech acts. This body of work depends fundamentally on agent communication languages, which are standardized sets of speech acts that ensure interoperability in agent to agent communication. The two most well known standards, KQML [5] and FIPA/ACL [6], are both based on speech act theory. FIPA/ACL also has a logical semantics defined using multi-modal BDI logic. But the semantics is meant only as a prescriptive guide when implementing software agents. Some researchers try to obtain, and sometimes even prove, conformance between the implementation and the semantics, while most programmers are probably not overly concerned with such matters. Moreover, the communication language is only a wrapper for a content language, which has to provide its own semantics. There is no integration of speech acts within a more general framework of action and change. Instead, these agent communication langauge technologies remain agnostic as to how to plan speech acts and other actions to achieve goals.

Morgenstern [16] offers an integrated theory of both types of actions using a *syntactic* first-order logic that includes quotation. Davis and Morgenstern [2] provide an alternative integration using regular first-order logic. The two theories' semantics cover both the speech acts and their content. However, while the theories were authored with the aim of applications in multi-agent planning, their use has so far been mainly of a prescriptive nature, in the implementation of a STRIPS-like planner in the case of the former theory, and as a specification for future implementations in the case of the latter theory.

In this paper we formalize inform and request speech acts in first-order logic with quotation. The representation is based on Temporal Action Logic (TAL), a first-order language with a well developed methodology for representing time, action, and change. TAL is complemented by syntactic operators that express the modalities of belief and commitment. They take quoted formulas as arguments and allow for the encoding of the effects of speech acts on other agents' beliefs and commitments. The resulting formalism

*This work is supported in part by the Swedish Foundation for Strategic Research (SSF) Strategic Research Center MOVIII, the Swedish Research Council Linnaeus Center CADICS, and CENIIT, the Center for Industrial Information Technology.

can be used to represent and reason about both speech acts and their message content, may it be facts, actions, or other speech acts. We automate such reasoning through a natural deduction theorem prover that incorporates a form of abductive planning. The system is applied to a multi-agent planning problem involving the cooperation between two robots through planned goal delegation and knowledge acquisition, which is introduced below.

2 Cooperation and Communication

Consider a motivating scenario involving an autonomous unmanned aerial vehicle named uav1. The robot is equipped with a winch system capable of lifting and dropping supply crates. Suppose it is assigned the task of delivering crate15 to the storage building store23. It would be unwise (although perhaps spectacular) to have the robot *fly into* the building. Instead, UAVs are restricted to operate in designated fly-zones, and storage buildings are not among them.

A class of autonomous unmanned ground vehicles provide services complementary to flying robots. They too can attach crates, using fork lifts, but stick to driving short distances in and between buildings designated as drive-zones. One of the UGVs, named ugv3, happens to sit idle in the building store14.

To succeed at its task the UAV will have to request help from the ground robot to get crate15 into the building, where it can not fly itself. It knows that ground robots have the capability of delivering crates between locations in drive-zones, and it might consider delegating its task to ugv3. But crate15's current location prevents simply delegating the goal since the crate is far outside any drive-zone areas where a ground vehicle could fetch it. Instead, uav1 will have to deliver crate15 to a rendezvous point, accessible to both UAVs and UGVs. Only then is it possible to request ugv3 to see to it that the crate gets to its final destination.

Such a plan is only possible if the two robots manage to coordinate their actions through communication. We would like them to figure out the above plan, including both physical actions and communicative speech acts, completely autonomously. This will require a sufficiently expressive representation and reasoning formalism. We present our proposal next.

3 Temporal Action Logic

First-order logic might serve as a solid foundation. But it is by itself too noncommittal regarding choices of how to represent actions and their effects on time-varying properties of the world. Several alternative logics of action and change are available to aid a logicist researcher. We present work with one such logic, the Temporal Action Logic (TAL).

The origins of TAL are found in Sandewall's model-theoretic Features and Fluents framework [23]. Doherty [3] selected important concepts, such as an explicit time line and the use of occlusion (discussed below), to form TAL and gave it a proof-theoretic first-order characterization. Many extensions since have turned TAL into a very expressive language for commonsense reasoning. Doherty and Kvarnström [4] provide a detailed account of the logic, but the version presented below includes further extensions that make TAL suitable for applications in multi-agent planning and reasoning.

In TAL, properties and relations that may change over time are modeled by *fluents*. A fluent f is a function of time, and its value at a time point t is denoted (value t f). When we talk about a time interval i between two time points t_1 and t_2 we mean the interval (t_1,t_2] that is open on the left and closed on the right. The functions (start i) and (finish i) picks out t_1 and t_2 respectively. An agent carrying out an action a during time interval i is specified by the predicate (Occurs *agent* i a). But the most important feature of TAL is probably its *occlusion* concept. A persistent fluent's value is permitted to change when occluded, but must persist during time intervals when not occluded. The following formula (with free variables implicitly universally quantified and in prefix form to make the representation of quoted formulas more convenient) relates a fluent f's value at the start and end time points of a time interval i:

$$(\rightarrow (\neg \text{(Occlude } i\ f)) \\ (= \text{(value (start } i) f) \text{ (value (finish } i) f))) \quad (1)$$

By assuming that fluents are not occluded unless otherwise specified one is in effect making the frame assumption that things usually do not change. Exceptions are specified by action specifications that explicitly occlude fluents that the action affects. E.g., if uav1 flies between two locations, its location fluent (location uav1) would be occluded during any interval with a non-empty intersection with the movement interval. This prevents any use of Formula 1 for relying on the default persistence of the robot's location that conflicts with the robot's moving about. By exercising fine-grained control over occlusion one gains a flexible tool for dealing with important aspects and generalizations of the frame problem.

3.1 A Syntactic Belief Operator

Previous accounts of TAL lack a representation of agents' mental states and beliefs. Introducing a *syntactic* belief operator provides a simple and intuitive notion of beliefs. To explain this let us first assume that uav1 believes it is at loc1 at noon. The following formula[1] would represent this belief in its knowledge base:

$$(= \text{(value 12:00 (location uav1))} \text{ loc1}) \quad (2)$$

[1] Clock times such as 12:00 are not really part of the logic. We assume a translation scheme between clock times and integers.

Similarly, if it was ugv3 that believed that uav1 is at loc1, Formula 2 would be in *its* knowledge base. Beliefs about others' beliefs are then really beliefs about what formulas are present in others' knowledge bases. If uav1 believes that ugv3 believes what Formula 2 expresses, then uav1 believes that ugv3 has Formula 2 in its knowledge base. This would be represented in the knowledge base of uav1 by the following formula:

(Believes ugv3 12:00
  '(= (value 12:00 (location uav1)) loc1))

The first argument of the Believes predicate is then the agent holding the belief. The second argument is the time point at which the agent holds the belief. Finally, the third argument is a quoted version of the formula expressing the belief, in this case Formula 2. This is what makes Believes a syntactic operator.

We use the quotation notation from KIF [7], which is a formal variant of Lisp's. An expression preceded by a quote is a regular first-order term that serves as a *name* of that expression. Alternatively one may use a backquote, in which case sub-expressions can be *unquoted* by preceding them with a comma. This facilitates *quantifying-in* by exposing chosen variables inside a backquoted expression for binding by quantifiers. E.g., we use quantifying-in to represent uav1's belief that ugv3 knows its own location, without uav1 having to know the name for that location:

(∃ x (Believes ugv3 12:00
    '(= (value 12:00 (location ugv3)) ',x)))

Note that while x ranges over locations[2], it is the *name* of a location that should occur as part of the third argument of Believes. The quote preceding the comma ensures that whatever value x is bound to is quoted to produce the name of that value.

While a quoted formula still looks like a formula, it is in fact a term. This means that standard inference rules such as modus ponens are not applicable to the quoted formulas that appear as arguments in the Believes operator. There are two possible solutions to this limitation. Either we could add axioms that express inference rules for beliefs, or we could employ a theorem prover with special purpose inference rules for beliefs. We pursue the latter alternative in the theorem prover described in Section 5, for efficiency reasons. While it should still be possible to characterize these inference rules in terms of axioms, this is subject to future work.

3.2 Action Occurrences

An action occurs when it is *possible* for an agent to execute the action, during some time interval i, and the agent is *committed* to the action occurring, at the start of the time interval. The predicate (Possible *agent* i *action*) represents physical and knowledge preconditions for an agent carrying out an action during time interval i, while (Committed *agent* t p) represents an agent's commitment at time point t to satisfy the formula p. Both predicates require a quoted expression in their third argument position, which precludes the free use of substitution of equals without regards to the agent's knowledge. Using these predicates we can formalize the above intuition about action occurrences:

(→ (∧ (Possible *agent* i ‘',*action*)
      (Committed *agent* (start i)
        ‘(Occurs ',*agent* ',i ',*action*)))
   (Occurs *agent* i *action*))

Note the interaction between backquote and quote in ‘',*action* to make sure that the argument of Possible is the *name* of the action. The initial backquote turns the following quote into the name of a quote, leaving the variable *action* free for binding. The resulting expression denotes the quoted version of whatever the variable is bound to rather than a quoted variable that can not be bound at all.

3.3 Action Specifications

Each one of an agent's available actions has an action specification that consists of three parts. The first part determines under what conditions an action is possible. It may include physical preconditions, but also involves knowledge preconditions on behalf of the agent executing the action.

Consider e.g. a stock market agent that plans to get rich by buying "the stock that will increase in value". While theoretically correct, the plan is of no practical value unless the agent knows a name that identifies some particular stock that it reasonably expects will increase in value. To make this intuition formal, Moore [14] suggests that an action is only executable if the agent knows *rigid designators* for all of the action's arguments. Morgenstern [15] modifies this suggestion slightly in her requirement that *standard identifiers* are known for the action arguments.

Our action specifications follow Morgenstern and use the syntactic predicate (Identifier x) to single out a name x as a standard identifier. In the stock market example, the action of buying stock would not be executable unless the agent in fact knew the name under which the stock was listed.

The second part of an action specification lists additional requirements for any agent that decides to execute the action itself. To execute an action the agent must invoke some piece of computer code implementing it. Since our actions have an explicit time argument we think of the agent as having an execution schedule to which procedure calls can be added at specific time points. Executing an action then involves looking up standard identifiers for its arguments and scheduling the procedure call associated with the action. The effect of deciding to execute an action is that the agent becomes committed to the action occurrence. An alternative way of ensuring commitment to an action is to delegate

[2] TAL is an order sorted logic. In our implementation we indicate variable sorts by prefixes, but ignore these here for readability.

its execution to someone else through the use of the request speech act, as we will see later.

Finally, the third part of an action specification details the effects of the action on the world and on the mental states of agents. This allows agents to reason about actions and form plans to achieve goals.

4 Formalization

We are now able to formalize the agent cooperation scenario presented in Section 2 using TAL. The following unique names assumptions are needed:

$(\neq l_1\ l_2)$ $\quad l_1, l_2 \in \{\text{base, helipad2, store14, store23}\}$
$(\neq c_1\ c_2)$ $\quad c_1, c_2 \in \{\text{nil, crate15}\}$

The terms in the first set are locations and the second are crates, where nil denotes a null element of the crate sort. In addition, quoted expressions are considered equal only when they are syntactically identical.

The term names in the following set are standard identifiers that can be used as arguments to procedure calls in the robot's internal action execution mechanism. We might e.g. imagine that the procedure for flying to one of the named locations involves a simple lookup of a GPS coordinate in an internal map data structure.

(Identifier x) $\quad x \in \{$'uav1, 'ugv3, 'crate15, 'base, 'helipad2, 'store14, 'store23$\}$

Operating restrictions on UAVs and UGVs are given by fly- and drive-zones:

(FlyZone base)
(FlyZone helipad2)
(DriveZone store14)
(DriveZone store23)
(DriveZone helipad2)

The above knowledge is common to all agents in our scenario.

4.1 Physical Actions

The bulk of the robots' knowledge base is made up of the action specifications. Each of the three specification parts are given by an implication. Starting with the fly action we note that it is possible for a UAV to fly to a location in a fly-zone if the UAV knows a standard identifier for the location. Secondly, an agent may commit to flying by scheduling a fly procedure call. The constant *self* is a placeholder for the identifier of the agent in whose KB the formula appears, e.g. uav1 or ugv3 in our case. This means that, while an agent can reason about whether it is possible for another agent to fly, it can not schedule a call to the fly procedure in another agent's execution mechanism. Thirdly, at the end of the fly interval the UAV ends up at its destination.

In addition, modified fluents need to be occluded to overrule their default persistence. Flying should occlude the UAV's location fluent in any interval that intersects the flying action since (Occlude i f) means that f is occluded *somewhere* in interval i. This could be expressed as an implication (→ (Intersect i_2 i) (Occlude i_2 (location *uav*))). However, the action specification below uses the contrapositive form of this formula. The reason for this is discussed further in Section 5.

(→ (∧ (Believes *uav* (start i) '(= ',*to* ,x))
(Identifier x)
(FlyZone *to*))
(Possible *uav* i '(fly ',*to*)))
(→ (∧ (= *to* x) (Identifier ',x)
(Schedule self i (fly x)))
(Committed self (start i) '(Occurs self ',i (fly ',*to*))))
(→ (Occurs *uav* i (fly *to*))
(∧ (= (value (finish i) (location *uav*)) *to*)
(→ (¬ (Occlude i_2 (location *uav*))) (Disjoint i_2 i))))

Note that the above might sometimes require the agent to reason about its own beliefs. Suppose, for example, that uav1 is considering the possibility of flying *itself* to ugv3's location. Its knowledge base might contain the formula (= (value t_1 (location ugv3)) helipad2), expressing the belief that ugv3 is at helipad2. Then uav1 would make the belief explicit by asserting (Believes uav1 t_2 '(= (value t_1 (location ugv3)) helipad2)), where t_2 is the current time.

Ground vehicles have a very similar action that allows them to drive to locations in drive-zones:

(→ (∧ (Believes *ugv* (start i) '(= ',*to* ,x))
(Identifier x)
(DriveZone *to*))
(Possible *ugv* i '(drive ',*to*)))
(→ (∧ (= *to* x) (Identifier ',x)
(Schedule self i (drive x)))
(Committed self (start i) '(Occurs self ',i (drive ',*to*))))

(→ (Occurs *ugv* i (drive *to*))
(∧ (= (value (finish i) (location *ugv*)) *to*)
(→ (¬ (Occlude i_2 (location *ugv*))) (Disjoint i_2 i))))

Both types of agents can carry one crate at a time, and the fluent (carrying *agent*) indicates which one it is at the moment. To attach a crate the agent must not already be carrying anything, indicated by the value nil, and the agent and crate must be at the same place. The action effects occlude the crate's location (as well as the carrying fluent) since we can no longer depend on the frame assumption that it will remain in the same place.

(→ (∧ (Believes *agent* (start i) '(= ',*crate* ,x))
(Identifier x)
(= (value (start i) (carrying *agent*)) nil)
(= (value (start i) (location *agent*))
(value (start i) (location *crate*))))
(Possible *agent* i '(attach ',*crate*)))

```
(→ (∧ (= crate x) (Identifier ‘’,x)
      (Schedule self i (attach x)))
   (Committed self (start i)
     ‘(Occurs self ’,i (attach ’,crate))))
(→ (Occurs agent i (attach crate))
   (∧ (= (value (finish i) (carrying agent)) crate)
      (→ (¬ (Occlude i2 (location crate)))
         (Disjoint i2 i))
      (→ (¬ (Occlude i3 (carrying agent)))
         (Disjoint i3 i))))
```

Detaching a crate has the effect that the crate ends up at the same location that the agent is currently at:

```
(→ (∧ (Believes agent (start i) ‘(= ’,crate ,x))
      (Identifier x)
      (= (value (start i) (carrying agent)) crate))
   (Possible agent i ‘(detach ’,crate)))
(→ (∧ (= crate x) (Identifier ‘’,x)
      (Schedule self i (detach x)))
   (Committed self (start i)
     ‘(Occurs self ’,i (detach ’,crate))))
(→ (Occurs agent i (detach crate))
   (∧ (= (value (finish i) (carrying agent)) nil)
      (= (value (finish i) (location crate))
         (value (finish i) (location agent)))
      (→ (¬ (Occlude i2 (location crate)))
         (Disjoint i2 i))
      (→ (¬ (Occlude i3 (carrying agent)))
         (Disjoint i3 i))))
```

4.2 Speech Acts

Speech acts can be used to communicate knowledge to, and to incur commitment in, other agents. We reformulate Allen's speech acts [1] in TAL using the syntactic belief and commitment predicates. More complex formulations have been suggested in the literature, e.g. to allow indirect speech acts [18]. But our robots will stick to straight answers and direct requests, without regard for politeness (although see Section 7 for a discussion of this).

The type of information we will be interested in is *knowing what* a particular value is. This is straight forwardly communicated by standard identifiers. E.g., if ugv3 wishes to inform uav1 that its location is store14 at noon, it may schedule an action of the following form:

```
(inform uav1
  ’(= (value 12:00 (location ugv3)) store14))          (3)
```

However, this is complicated when uav1 wishes to *ask* ugv3 what its location is. In accordance with much research in speech acts, we view questions as requests for information. The UAV should thus request that the UGV perform the inform action in Formula 3. Though since uav1 does not know where ugv3 is, which is presumably the reason why it is asking about it in the first place, it can not know what action to request.

Again we follow Allen's directions and introduce an informRef action designed to facilitate questions of this type. The informRef action does not mention the value that is unknown to the UAV agent, which instead performs the following request:

```
(request ugv3
  ’(Occurs ugv3 i2 (informRef uav1
                     (value 12:00 (location ugv3)))))
```

The above request still contains the unknown time interval i2, which ugv3 may instantiate in any way it chooses. The explicit time representation used by TAL opens up the possibility of a general account of the knowledge preconditions and knowledge effects of action's start and end time points, but formulating it is part of future work.

The informRef preconditions require that the informing agent *knows what* the value is, which is being informed about. The effects assert the existence of a value for which the speaker knows a standard name.

Note that an agent that commits to *executing* the action schedules an *inform* procedure call, plugging in the sought value. In contrast, an agent that only *reasons* about the effects of the informRef action, as in the question example above, knows that the value will become known, but need not yet know its name.

```
(→ (∧ (Believes speaker (start i) ‘(= ’,value ,x))
      (Identifier x)
      (Believes speaker (start i) ‘(= ’,hearer ,y))
      (Identifier y))
   (Possible speaker i ‘(informRef ’,hearer ’,value)))
(→ (∧ (= value x) (Identifier ‘’,x)
      (= hearer y) (Identifier ‘’,y)
      (Schedule self i (inform y ‘(= ’,value ’,x))))
   (Committed self (start i)
     ‘(Occurs self ’,i (informRef ’,hearer ’,value))))
(→ (Occurs speaker i (informRef hearer value))
   (∃ x (∧ (Believes hearer (finish i) ‘(= ’,value ,x))
           (Identifier x))))
```

Many other formalizations of speech acts restrict requests to action occurrences. Our formulation of requests supports any well formed formulas, whether they are declarative goals or action occurrences. The effect is that the agent that is the target of the request is committed to satisfying the formula.

```
(→ (∧ (Wff formula)
      (Believes speaker (start i) ‘(= ’,hearer ,x))
      (Identifier x))
   (Possible speaker i ‘(request ’,hearer ’,formula)))
(→ (∧ (Wff formula)
      (= hearer x) (Identifier ‘’,x)
      (Schedule self i (request x formula)))
   (Committed self (start i)
     ‘(Occurs self ’,i (request ’,hearer ’,formula))))
(→ (Occurs speaker i (request hearer formula))
   (Committed hearer (finish i) formula))
```

The Wff predicate determines whether the quoted expression is a well formed formula. While we could write axioms defining it, since quoted expressions are terms in our language, we find it convenient to view it as defined by semantic attachment.

Finally, to delegate declarative goals an agent must know something about the capabilities of other agents. In our scenario, UAVs know that ground robots are able to transport crates between locations in drive-zones. This allows uav1 to delegate its goal task and trust that it will indeed be satisfied.

(→ (∧ (DriveZone (value (start i) (location *crate*)))
(DriveZone *to*)
(Committed *ugv* (start i)
'(= (value (finish ',i) (location ',*crate*)) ',*to*)))
(= (value (finish i) (location *crate*)) *to*))

This concludes our formalization of the robot cooperation scenario. We turn our attention next towards the question of how to perform automated reasoning with it.

5 Automated Natural Deduction

Earlier work with TAL has made use of a model-theoretic tool for automated reasoning called VITAL [9]. But this tool relies upon the set of actions being pre-specified and consequently does not support planning. Later work made deductive planning possible through a compilation of TAL formulas into Prolog programs [10]. But Prolog's limited expressivity makes it inapplicable to interesting planning problems involving incomplete information and knowledge producing actions, such as speech acts. Instead, our current work concentrates on an implementation of a theorem prover based on *natural deduction*, inspired by similar systems by Rips [22] and Pollock [20].

Natural deduction is an interesting alternative to the widely used resolution theorem proving technique. A natural deduction prover works with the formulas of an agent's knowledge base in their "natural form" directly, rather than compiling them into clause form. The set of proof rules is extensible and easily accommodates special purpose rules that can make reasoning in specific domains or using a specific formalism like TAL more efficient. We are actively experimenting with different rule sets so the description below is of a preliminary nature.

Rules are divided into *forward* and *backward* rules. Forward rules are applied whenever possible and are designed to converge on a stable set of conclusions so as not to continue generating new inferences forever. Backward rules, in contrast, are used to search backwards from the current proof goal and thus exhibits goal direction. Combined, the result is a bi-directional search for proofs.

Nonmonotonic reasoning and planning is made possible in our theorem prover through an assumption-based argumentation system. The set of *abducibles* consists of negated occlusion, action occurrences, temporal constraints, and positive or negative holds formulas, depending on the current reasoning task [13]. These are allowed to be assumed rather than proven, as long as they are not counter-explained or inconsistent. As an example, consider the following natural deduction proof fragment, explained below (where the justifications in the right margin denote row numbers, (P)remises, (H)ypotheses, and additional background (K)nowledge).

1	(= (value 12:00 (location uav1)) base)	P
2	(∧ (= (start i37) 12:00) (= (finish i37) 13:00))	P
3	┌ (¬ (Occlude i37 (location uav1)))	H
4	│ (= (value 13:00 (location uav1)) base)	1-3,K
5	│ (= helipad2 helipad2)	K
6	│ (Believes uav1 (start i38) '(= helipad2 helipad2))	5
7	│ (Possible uav1 i38 '(fly helipad2))	6,K
8	│ ┌ (Schedule uav1 i38 (fly helipad2))	H
9	│ │ (Committed uav1 (start i38) '(Occurs uav1 i38 (fly helipad2)))	8,K
10	│ │ (Occurs uav1 i38 (fly helipad2))	7,9,K
11	│ │ (= (value (finish i38) (location uav1)) helipad2)	10,K
12	│ │ (→ (¬ (Occlude i (location uav1))) (Disjoint i i38))	10,K
13	│ │ (Disjoint i37 i38)	3,12

The UAV is located at base at noon, as in Row 1. Suppose it needs to remain at the same location at 1 p.m. One way of proving this would be by using the persistence formula in Section 3. The location fluent is only persistent if it is not occluded. While uav1 has no knowledge about whether it is occluded or not, (¬ Occlude) is abducible and may be (tentatively) *assumed*. The effect of making non-occlusion abducible is to implement a default persistence assumption. Row 2 introduces a fresh interval constant and Row 3 indicates the assumption using a Copi style (described e.g. by Pelletier [17]) vertical line in the margin.

Suppose further that uav1 also needs to visit helipad2. The only way of proving this would be to use the fly action defined in Section 4. A backward modus ponens rule adopts (Occurs uav1 i38 (fly helipad2)) as a sub goal. Backward chaining again, on the action occurrence axiom in Section 3, causes (Possible uav1 i38 '(fly helipad2)) and (Committed uav1 (start i38) '(Occurs uav1 i38 (fly helipad2))) to become new sub goals. These are again specified by the fly action specification. The first of these sub goals is satisfied by Row 6 and the fact that helipad2 is both an identifier and a fly-zone. The commitment goal in Row 9 is satisfied by Row 5, the fact that helipad2 is a viable argument to the fly procedure, and Row 8, which assumes that uav1 schedules the procedure call. The implementation of the proof rule that adds Row 8 performs the actual scheduling by updating an internal data structure. It is still possible to backtrack, removing the assumption in Row 8, as long as the procedure call has not yet been executed, i.e. if it is scheduled to occur at some future time or if execution has not yet reached this

point. This could happen if something causes the theorem prover to reconsider flying to helipad2, or if scheduling the flight causes a conflict with some other assumption that was made previously. In such cases the procedure call would be removed from the internal data structure as well.[3] Finally, having proved the robot's ability and commitment to flying to helipad2 Row 10 concludes that the flight will occur, with the effect that uav1 ends up at helipad2 in Row 11.

Flying should occlude the location fluent in any intersecting interval. This would most naturally be expressed by (→ (Intersect i i38) (Occlude i (location uav1))). But, as noted in Section 4, we use the contrapositive form instead. The reason is the need for consistency checking when assumptions have been made. It is well known that the problem of determining consistency of a first-order theory is not even semi-decidable. Our theorem prover uses its forward rules to implement an *incomplete* consistency check (more on this below), and the contrapositive form makes these forward rules applicable. Row 12, which is an effect of the fly action, together with the assumption in Row 3 trigger the forward modus ponens rule, adding the disjointness constraint in Row 13. This enforces a partial ordering of the two intervals to avoid any conflict between the persistence of the UAV's location, and its moving about. Another forward inference rule consists of a constraint solver that determines whether the set of temporal constraints is consistent. If it is impossible to order i37 and i38 so that they do not intersect in any way, then an inconsistency has been detected and the prover needs to backtrack, perhaps cancelling the most recent assumption or removing the action that was last added to the schedule.

For some restrictions on the input theory we are able to guarantee completeness of the nonmonotonic reasoning [13]. But in the general case, when one cannot guarantee completeness of the consistency checking, we might conceivably fail to discover that one of the assumptions is unreasonable. This would not be a cause of unsoundness, since we are still within the sound system of natural deduction, but it might result in plans and conclusions that rest on impossible assumptions. A conclusion Φ depending on an inconsistent assumption would in effect have the logical form $\perp \rightarrow \Phi$, and thus be tautological and void. This is to be expected though. Since consistency is not even semi-decidable, the most one can hope for is for the agent to continually evaluate the consistency of its assumptions, improving the chances of them being correct over time, while regarding conclusions as tentative. [21].

6 Generated Plans

By applying the natural deduction theorem prover to the TAL formalization we are able to automatically generate plans for the robot cooperation scenario. We present the proof goals and the resulting plans below.

Let us initially place the crate and the UAV (carrying nothing) at base at 12:00:

(⊨ (value 12:00 (location crate15)) base)
(⊨ (value 12:00 (location uav1)) base)
(⊨ (value 12:00 (carrying uav1)) nil)

The goal is to have crate15 delivered to the storage named store23 at some future time point:

Show (∃ t (⊨ (value t (location crate15)) store23))

The UAV uses theorem proving to reason backwards from this goal approximately like what follows. "For the crate to be at store23 someone must have put it there. I could put it there myself if I was located at store23 carrying crate15. But I can't think of any way to satisfy the fly-zone precondition of flying to store23. Though my knowledge of ground vehicles suggests a completely different possibility. My goal would also be satisfied if both the crate's location and store23 were in drive-zones, and some ground vehicle had committed to the goal. In fact, helipad2 is a drive-zone, and it is also a fly-zone, so I can go there and drop the crate off. Before going there I should attach crate15, which is right here next to me. Then I'll decided upon some particular ground robot, say, ugv3, and request that it adopts the goal that crate15 is at store23."

While the robots are not nearly as self aware as this monologue suggests, it corresponds roughly to the search space for the following plan:

```
(Schedule uav1 i1 (attach crate15))
(Schedule uav1 i2 (fly helipad2))
(Schedule uav1 i3 (detach crate15))
(Schedule uav1 i4
  (request ugv3
    '(⊨ (value (finish i5) (location crate15)) store23)))
(Before i1 i2)
(Before i2 i3)
(Before i3 i4)
(Before i4 i5)
```

The UAV executes its plan, including sending the goal request to ugv3. We switch to look inside the mind of the UGV as it tries to prove that the requested formula is satisfied. Suppose that half an hour has passed and that the UGV happens to be at some other storage building, carrying nothing:

(⊨ (value 12:30 (location ugv3)) store14)
(⊨ (value 12:30 (carrying ugv3)) nil)

The UGV will have to drive to the crate in order to pick it up and deliver it to store23. But ugv3 does not know crate15's location, and scheduling a drive to (value 12:30 (location crate15)) is prevented by the Identifier requirement on the drive action argument. The restriction is necessary since trying to find the coordinate of (value 12:30 (location crate15)) will certainly not generate any results given

[3] The link between theorem proving and action execution is an interesting topic. The mechanism described above is one approach, but we are currently investigating alternatives.

the robot's area map. The plan should instead involve finding a standard identifier for crate15's current location and looking *that* up in the map.

We assume that ugv3 believes that uav1 knows where crate15 is, and that whatever location that is, it is a drive-zone (although see Section 7 for a discussion of this):

```
(∃ x (∧ (Believes uav1 12:30
           '(= (value 12:30 (location crate15)) ,x))
        (Identifier x)))
(DriveZone (value 12:30 (location crate15)))
```

The task is then to prove the content of uav1's request:

```
Show (= (value (finish i5) (location crate15)) store23)
```

The resulting plan makes use of the request and informRef speech act combination to formulate a question corresponding to "what is crate15's location". Furthermore, while this question will equip the robot with a standard identifier, this identifier is not yet known at the time the plan is being constructed. Rather than scheduling the drive procedure call, ugv3 instead plans to *request itself* to carry out the driving after having asked uav1 about crate15's location. At the time at which this request is managed, the required information will be available for scheduling the actual drive procedure call. The rest should be a simple matter of going to store23 to drop crate15 off at its goal:

```
(Schedule ugv3 i6
  (request uav1
    '(Occurs uav1 i7
       (informRef ugv3
         (value (start i5) (location crate15))))))
(Schedule ugv3 i8
  (request ugv3
    '(Occurs ugv3 i9
       (drive (value (start i5) (location crate15))))))
(Schedule ugv3 i10 (attach crate15))
(Schedule ugv3 i11 (drive store23))
(Schedule ugv3 i12 (detach crate15))
(Before i6 i7)
(Before i7 i8)
(Before i8 i9)
(Before i9 i10)
(Before i10 i11)
(Before i11 i12)
```

Let us switch our attention back to uav1 and see what it plans to do about ugv3's request for information. The UAV's current state is described by:

```
(= (value 12:30 (location crate15)) helipad2)
(= (value 12:30 (location uav1)) helipad2)
(= (value 12:30 (carrying uav1)) nil)
```

The proof goal is defined by the incoming request:

```
Show (Occurs uav1 i7
        (informRef ugv3
          (value (start i5) (location crate15))))
```

Since uav1 has first hand knowledge about crate15's location it schedules an inform procedure call according to the definition of the informRef speech act:

```
(Schedule uav1 i7
  (inform ugv3
    '(= (value (start i5) (location crate15)) helipad2)))
```

Switching our focus back to ugv3 we find that it has received the formula that uav1 informed it about:

```
(= (value (start i5) (location crate15)) helipad2)
```

This puts ugv3 in a position where it can prove the content of its request to itself:

```
Show (Occurs ugv3 i9
        (drive (value (start i5) (location crate15))))
```

The result is that the missing drive procedure call is inserted at the right place in the execution schedule with the standard identifier plugged in as its argument:

```
(Schedule ugv3 i9 (drive helipad2))
```

Once at helipad2, the rest of the scheduled actions will have the robot attaching crate15, driving to store23, and dropping the crate off to satisfy the goal and complete the scenario.

7 Limitations and Future Work

The work presented in this paper is far from a complete solution to the robot cooperation scenario. One unsolved question regards our assumption that ugv3 believes that uav1 knows where crate15 is. Maybe there ought to be some commonsense knowledge that would allow it to defeasibly infer uav1's knowledge from the fact that it delegated a goal that directly involved that knowledge. One might suspect that this is but one instance of a more general problem of reasoning about who is likely to know what in which situations. An alternative solution would be to have the UAV reason about the fact that ugv3 needs to know where the crate is to be able to move it to its destination. The UAV could then pro-actively inform the UGV about the crate's location before requesting the UGV to move it.

An ad hoc move that we were forced to make was to remove the informRef speech act from the UAVs knowledge base while generating the first plan. While this particular action is not needed for that particular plan, the UAV clearly ought to have access to all its actions at all times. The reason for our move has to do with the fact that uav1 must attempt to solve the goal itself before considering delegating it. What makes our scenario interesting is that it is not possible to solve without cooperation. But uav1 can not know that trying to deliver crate15 by itself is futile until it has explored all alternative ways of doing so. Unfortunately, the informRef speech act made for a rather unwieldy search space, which was more than our theorem prover had time to explore while we cared to wait. This prevented

uav1 from giving up on the prospect of managing the delivery by itself within a reasonable amount of time. We suspect that as the agents are equipped with more knowledge and actions, more possibilities will open up in the theorem prover's search space, and the need for some kind of heuristic to help guide search will increase.

The speech acts themselves are subject to some limitations. One is our disregard of any physical preconditions to communication such as geographical closeness constraints. Our robots are assumed to have a radio link at all times. Another limitation is that we do not consider indirect speech acts. This seems reasonable as long as we are thinking of communication between our robots. But there is no denying that many of the speech acts uttered by *humans* are indirect. A human UAV operator uttering "Could you make sure that crate15 is in store23?" is presumably requesting the UAV to make sure the goal is satisfied rather than querying about its ability to do so. Another serious limitation is our assumption that other agents always accept requests. Some rejected requests could reasonably be handled during plan execution through re-planning or plan repair. But others should be considered already during planning and would result in conditionally branching plans or plans with loops that repeat requests until accepted.

A future development could be the inclusion of composite actions, which would make it possible to explicitly represent informRef as a macro action that includes an inform speech act. This is in contrast to our current formalization where inform is only a procedure call and not a stand alone action. Another possibility for development exists with regards to the execution schedule mechanism. While we think that it is a promising method for integrating planning and execution, the description of its workings that we have provided here is rather sketchy and needs further elaboration. In particular we would like to take advantage of our integrated temporal constraint solver to calculate action durations and schedule actions at explicit clock times.

Finally, an agent architecture based exclusively on logical reasoning raises efficiency concerns. Both plans in our running example were automatically generated by the theorem prover in 2 minutes and 35 seconds on a Pentium M 1.8 GHz laptop with 512 MB of RAM. That might or might not be reasonable, depending on the application. But, in either case, it was admittedly a small problem, which begs the question of whether the architecture will scale up to real-world problems. Alas, we do not yet know. But there are at least some reasons to be optimistic.

One reason is, as already mentioned, the use of a temporal constraint solver for reasoning with time. More generally, one can view special purpose algorithms as additional natural deduction rules that make certain types of inferences efficient. Another reason is the choice of an interruptible algorithm for nonmonotonic reasoning. In a real-time setting the agent can act, at any time, to the best of its knowledge given the reasoning it has performed up to that point.

But most encouragingly, achieving satisfactory performance in certain domains is already possible. E.g., our theorem prover was applied to UAV surveillance and quickly generated plans for realistic size problems [11]. Furthermore, the agent architecture was used to the control the characters in a computer game that requires real-time interaction [12]. We believe computer games to be a particularly suitable domain for empirical studies of logical agents on the road from tiny benchmark problems towards larger real-world applications.

8 Conclusions

We have described a scenario involving communication and cooperation between two robots. The solution required one robot to plan to delegate a goal through communication using a request speech act. The other robot had to plan to achieve knowledge preconditions, again through communication using a nested request and informRef speech act. These speech acts were formalized in an extension of Temporal Action Logic that includes syntactic belief and commitment operators, which were made possible through the use of a quotation mechanism. The formalization made it possible to generate a plan involving both cooperation and communication using automated theorem proving. Finally, a novel scheduling mechanism provided a tightly coupled integration between planning and the execution of generated plans.

The formalization used quotation, which seems most befitting of a logicist framework. The robots' explicit representation of beliefs as formulas in a knowledge base motivates their representation of others' beliefs as quoted formulas. Further benefits may be gained by using quotation in the context of speech acts. A fuller theory of communication will presumably also include locutionary acts, i.e. the actual utterances that encode messages between agents. These are most naturally thought of as strings consisting of quoted formulas from the agents' knowledge bases.

Our philosophy is based on the principle that logic is an intelligent agent's "language of thought". The formalization of the speech acts are similar to their corresponding semantics proposed in the literature. But unlike many other approaches that view the semantics as normative, such as agent communication languages, we put the formulas in our agents' heads where the agents can reason with them using theorem proving. In fact, our use of a prefix notation for formulas makes the correspondence between the theory in this paper and its Lisp implementation *exact*, save for some logical symbols that are not available for use as Lisp identifiers. Through this approach we hope to construct an agent architecture based on logical planning with a level of flexibility that would be difficult to match using agent programming languages.

References

[1] James Allen. *Natural Language Understanding*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1988.

[2] Ernest Davis and Leora Morgenstern. A first-order theory of communication and multi-agent plans. *Journal of Logic and Computation*, 15(5):701–749, 2005.

[3] Patrick Doherty. Reasoning about action and change using occlusion. In *Proceedings of the Eleventh European Conference on Artificial Intelligence ECAI'94*, pages 401–405, 1994.

[4] Patrick Doherty and Jonas Kvarnström. Temporal action logics. In Vladimir Lifschitz, Frank van Harmelen, and Bruce Porter, editors, *Handbook of Knowledge Representation*. Elsevier, 2007.

[5] Tim Finin, Jay Weber, Gio Wiederhold, Michael Genesereth, Richard Fritzson, Donald McKay, James McGuire, Richard Pelavin, Stuart Shapiro, and Chris Beck. Specification of the KQML agent-communication language. Technical Report EIT TR 92-04, Enterprise Integration Technologies, Palo Alto, CA, July 1993.

[6] Foundation for Intelligent Physical Agents. FIPA communicative act library specification. `http://www.fipa.org/specs/fipa00037/`, 2002.

[7] Michael R. Genesereth and Richard E. Fikes. Knowledge interchange format, version 3.0 reference manual. Technical Report Logic-92-1, Computer Science Department, Stanford University, June 1992.

[8] Michael R. Genesereth and Steven P. Ketchpel. Software agents. *Communications of the ACM*, 37(7):48–53, 1994.

[9] Jonas Kvarnström. VITAL: Visualization and implementation of temporal action logics. `http://www.ida.liu.se/~jonkv/vital/`, 2007.

[10] Martin Magnusson. *Deductive Planning and Composite Actions in Temporal Action Logic*. Licentiate thesis, Linköping University, September 2007. `http://www.martinmagnusson.com/publications/magnusson-2007-lic.pdf`.

[11] Martin Magnusson and Patrick Doherty. Deductive planning with inductive loops. In Gerhard Brewka and Jérôme Lang, editors, *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 528–534. AAAI Press, 2008.

[12] Martin Magnusson and Patrick Doherty. Logical agents for language and action. In Christian Darken and Michael Mateas, editors, *Proceedings of the 4th Artificial Intelligence and Interactive Digital Entertainment Conference AIIDE-08*. AAAI Press, 2008.

[13] Martin Magnusson, Jonas Kvarnström, and Patrick Doherty. Abductive reasoning with filtered circumscription. In *Proceedings of the 8th Workshop on Nonmonotonic Reasoning, Action and Change NRAC 2009*. UTSePress, 2009. Forthcoming.

[14] Robert Moore. Reasoning about knowledge and action. Technical Report 191, AI Center, SRI International, Menlo Park, CA, October 1980.

[15] Leora Morgenstern. Knowledge preconditions for actions and plans. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 867–874, 1987.

[16] Leora Morgenstern. *Foundations of a logic of knowledge, action, and communication*. PhD thesis, New York, NY, USA, 1988. Advisor: Ernest Davis.

[17] Francis Jeffry Pelletier. A brief history of natural deduction. *History and Philosophy of Logic*, 20:1–31, 1999.

[18] C. Raymond Perrault and James F. Allen. A plan-based analysis of indirect speech acts. *Computational Linguistics*, 6(3-4):167–182, 1980.

[19] C. Raymond Perrault, James F. Allen, and Philip R. Cohen. Speech acts as a basis for understanding dialogue coherence. In *Proceedings of the 1978 workshop on Theoretical issues in natural language processing*, pages 125–132, Morristown, NJ, USA, 1978. Association for Computational Linguistics.

[20] John Pollock. Natural deduction. Technical report, Department of Philosophy, University of Arizona, 1999. `http://www.sambabike.org/ftp/OSCAR-web-page/PAPERS/Natural-Deduction.pdf`.

[21] John L. Pollock. *Cognitive Carpentry: A Blueprint for how to Build a Person*. MIT Press, Cambridge, MA, USA, 1995.

[22] Lance J. Rips. *The psychology of proof: deductive reasoning in human thinking*. MIT Press, Cambridge, MA, USA, 1994.

[23] Erik Sandewall. *Features and Fluents: The Representation of Knowledge about Dynamical Systems*, volume 1. Oxford University Press, 1994.

[24] John R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

Active logic and practice

Jacek Malec
Department of Computer Science, Lund University
Box 118, 221 00 LUND, Sweden
jacek.malec@cs.lth.se

May 9, 2009

Abstract

The problem of finding a suitable formal approach to describe on-going reasoning process has been open since the very beginning of AI. In this paper we argue that active logic might be a formalism useful in this context. Active logic is first introduced, then we analyse resource limitations that constrain the space of possible practical realisations of such reasoners. Finally some steps towards creating a practical active logic reasoner are presented.

1 Introduction

The problem of finding a suitable formal approach to describe on-going reasoning process has been open since the very beginning of AI. In particular, the areas of reasoning about action and change, belief revision, defeasible reasoning, interleaving planning and acting in dynamic domains, have all addressed this problem, albeit partially, from different points of view. However, there is no well-developed theory of reasoning viewed as an activity performed in-time.

Another aspect of practical reasoning is that it is *always* performed with limited resources. Our approximations normally neglect this aspect, or address only some specific issue like real-time deadlines or limited memory footprint. But there is no formalism available yet that would provide a reliable starting point for building a reasoner able to tackle all the resource limitations occurring in practice.

One possibility, quite often adopted by practitioners, is to forget the theory and build systems that act irrespectively of the lack of appropriate formal grounds, suitable theory or complete explanation. There exist robots able to deal with apparently very complex dynamic environments (see e.g., results of the DARPA Urban Challenge [6]). The missing of theoretical grounds and incomplete formal reasoning, however apparent in many cases, do not preclude those systems from efficient, timely action. As a counterweight, an interesting attempt to base a practical system on well-founded grounds of formal reasoning, relevant in the context of this Workshop, is summarized in a recent thesis [14].

There have been numerous attempts to come up with theories of reasoning capable to be adapted to real-world constraints and limitations. We are not going to present them here but will focus on one particular approach worth reminding and, in our opinion, worth also further consideration. The paper will introduce active logic in the next section, then a short discussion of resource limitations will be presented. In the following section we will introduce our ongoing work in this area. Finally, some conclusions are stated.

2 Active Logic

The very first idea for our investigations [3] has been born from the naive hypothesis that in order to be able to use symbolic logical reasoning in a real-time system context it would be sufficient to limit the depth of reasoning to a given, predefined level. This way one would be able to guarantee predictability of a system using this particular approach to reasoning. Unfortunately, such a modification performed on a classical logical system yields a formalism with a heavily modified and, in princi-

ple, unknown semantics [18]. It would be necessary to relate it to the classical one in a thorough manner. This task seems very hard and it is unclear for us what techniques should be used to proceed along this line. But the very basic idea of "modified provability": *A formula is a theorem iff it is provable within n steps of reasoning*, is still appealing and will reappear in various disguises in our investigations.

The next observation made in the beginning of this work was that predictability (in the hard real-time sense) requires very tight control over the reasoning process. In the classical approach one specifies a number of axioms and a set of inference rules, and the entailed consequences are expected to "automagically" appear as results of an appropriate consequence relation. Unfortunately, this relation is very hard to compute and usually requires exponential algorithms. One possibility is to modify the consequence relation in such way that it becomes computable. However, the exact way of achieving that is far from obvious. We have investigated previous approaches and concluded that a reasonable technique for doing this would be to introduce a mechanism that would allow one to control the inference process. One such mechanism is available in Labeled Deductive Systems [12].

In its most simple, somewhat trivialized, setting a labeled deductive system (LDS) attaches a *label* to every well-formed formula and allows the inference rules to analyze and modify labels, or even trigger on specific conditions defined on the labels. E.g., instead of the classical Modus Ponens rule $\frac{A, A \to B}{B}$ a labeled deduction system would use $\frac{\alpha:A,\ \beta:A \to B}{\gamma:B}$, where α, β, γ belong to a well-defined language (or, even better, algebra defined over this language) of labels, and where γ would be an appropriate function of α and β. If we were to introduce our original idea of limited-depth inference, then γ could be, e.g., $\max(\alpha, \beta) + 1$ provided that α and β are smaller than some constant N.

A similar idea, although restricted to manipulation of labels which denote time points, has been introduced in *step-logic* [9] which later evolved into a family of *active logics* [11]. Such a restriction is actually a reasonable first step towards developing a formal system with provable computational properties. Active logics have been used so far to describe a variety of domains, like planning [17], epistemic reasoning [10], reasoning in the context of resource limitations [16] or modeling discourse. Quite recently there has been some successful work devoted to determining appropriate semantics for active logic systems [1]. However, only single-agent, static variant of the logic is covered there.

The real strength of active logic comes from the fact that labels are understood as discrete time points and that the set of premises used for reasoning may change with time. This way the formalism is prepared to accept fresh "observations" every "clock tick", thus extending the static logical consequence into the time dimension. Another very important aspect of active logic is its paraconsistency, together with some mechanisms allowing removal of contradicting formulae from the knowledge base. These two latter properties are crucial for modelling practical systems with resource limitations, see e.g. [15].

3 Practical Reasoner

Active logic implemented according to the original definitions referenced above unavoidably suffers from combinatorial explosion of the number of formulae associated with every time step. Locally, it is still a classical logical system. However, early in the work on active logic an idea of limited memory areas, somehow similar to human short-term memory, has been introduced [8]. The model is slightly more complex, with five memory banks fulfilling different functions (see Fig. 1). We have found it very appealing from the point of view of limited memory resources. In order to make it amenable for further analysis, we have formulated it as an LDS and tested its behaviour for hand, on some very simple examples [2, 4].

4 Paraconsistent Robot

In order to realise the idea of building a practical reasoner capable of taking account of time as it flows (in order to obey deadlines), capable of resolving inconsistencies in its knowledge base (due to e.g., erroneous observations corrected at some later point) and able to take into account its own limitations (memory size, processor speed or energy consumption) we have begun with a scenario involving

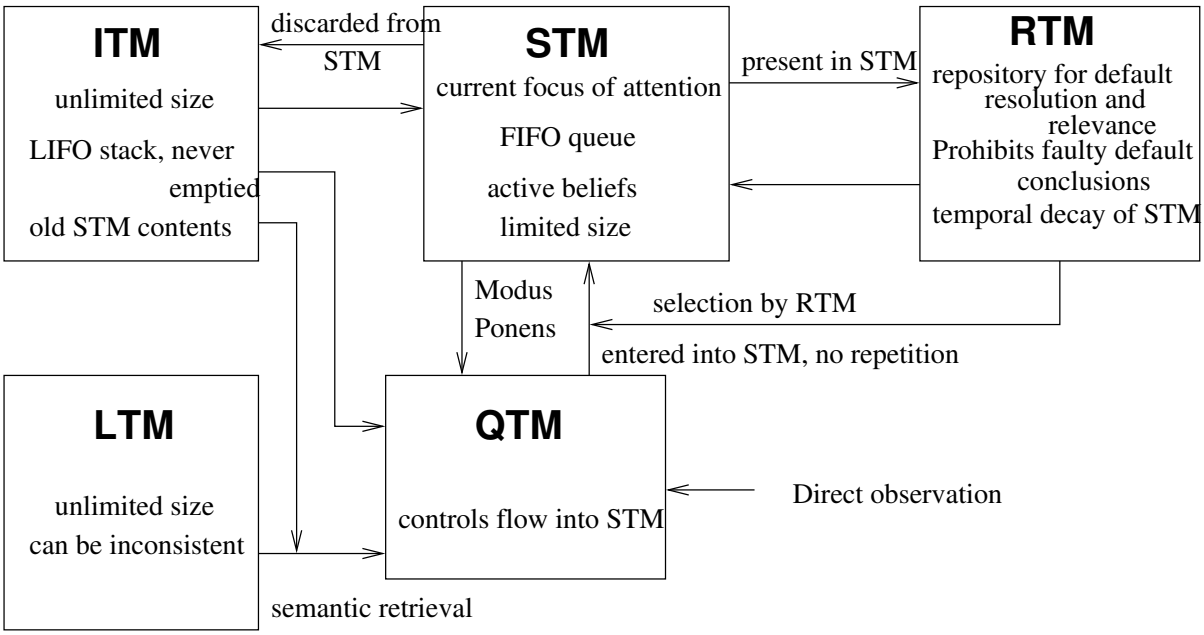


Figure 1: The memory model from [8].

a day of life of a service robot [13]. This scenario is loosely based on the much more interesting"Seven days in the life of a robotic agent" [5]. The idea is that a service robot has a series of tasks to be performed during the day, some of them more important than others and some provided with hard deadlines. A normal day plan for the robot allows the schedule (and all the deadlines) to be met. However, some day a problem occurs, requiring the robot to realise the problem and replan. The questions that might arise (in the rough order of complexity) are: What is the problem? Is the current plan inapplicable? Can I find a new plan to reach my goals? Can I find a plan meeting all the deadlines? Can I find one in time to meet deadlines (I can't reason too long then)? Can I find one given my current resources? e.t.c.

We have begun by creating a theorem prover capable to take an LDS specification (consistent with the formalisation from [2]). It has been shown to correctly prove a number of active logic theorems [7], in particular with observations coming as a stream of data while reasoning. Then we have applied it to our robot-day scenario [13], with the conclusion that in principle the prover is capable of performing the necessary reasoning, however it suffers from inefficient implementation and possible memory leaks. At this point we consider rewriting the prover again with speed as the major design objective.

5 Conclusions

In this paper we have briefly presented active logic and our work trying to apply it in scenarios relevant for continuous reasoning. In particular, active logic allows for reasoning in time, incorporating on the way an incoming stream of observations. It also lets us take care of inconsistencies. Embedding it in a mechanizable LDS allows us to take into account physical resource limitations, thus making the resulting system applicable in practice.

The original title of this paper was "Active Logic in Practice". However, I have realised that although this is my intention, it still requires a lot of both theoretical and practical effort to get to the point when we can say that active logic is usable in practice. In particular, we need to address the following points: efficiency of the prover; implementation on a robot, involving transforming its physical

stream of sensory data into a symbolic stream of observations; further theory development: what do we really implement? how can this model be extended onto multiple cooperating agents? Those, and many other, questions require a lot of work before we can say that a practically useful reasoning system has been developed.

Acknowledgements

The author is indebted to Mikael Asker and Johan Hovold for their work on active logic theory, and to Sonia Fabre, Julien Delnaye and Thorben Ole Heins for their work on theorem provers.

References

[1] M. L. Anderson, W. Gomaa, J. Grant, and D. Perlis. Active logic semantics for a single agent in a static world. *Artificial Intelligence*, 172:1045–1063, 2008.

[2] M. Asker. Logical reasoning with temporal constraints. Master's thesis, Department of Computer Science, Lund University, August 2003. Available at `http://fileadmin.cs.lth.se/ai/xj/-MikaelAsker/exjobb0820.ps`.

[3] M. Asker and J. Malec. Reasoning with limited resources: An LDS-based approach. In et al. B. Tessem, editor, *Proc. Eight Scandinavian Conference on Artificial Intelligence*, pages 13–24. IOS Press, 2003.

[4] M. Asker and J. Malec. Reasoning with limited resources: Active logics expressed as labelled deductive systems. *Bulletin of the Polish Academy of Sciences, Technical Sciences*, 53(1), 2005.

[5] W. Chong, M. O'Donovan-Anderson, Y. Okamoto, and D. Perlis. Seven days in the life of a robotic agent. Technical report, University of Maryland, 2002.

[6] DARPA. Urban Challenge. `http://www.darpa.mil/GRANDCHALLENGE/-overview.asp`, 2007.

[7] J. Delnaye. Automatic theorem proving in active logics. Master's thesis, Department of Computer Science, Lund University, June 2008. Available at `http://fileadmin.cs.lth.se/ai/xj/-JulienDelnaye/report.pdf`.

[8] J. Drapkin, M. Miller, and D. Perlis. A memory model for real-time commonsense reasoning. Technical Report TR-86-21, Department of Computer Science, University of Maryland, 1986.

[9] J. Elgot-Drapkin. *Step Logic: Reasoning Situated in Time*. PhD thesis, Department of Computer Science, University of Maryland, 1988.

[10] J. Elgot-Drapkin. Step-logic and the three-wise-men problem. In *Proc. AAAI*, pages 412–417, 1991.

[11] J. Elgot-Drapkin, S. Kraus, M. Miller, M. Nirkhe, and D. Perlis. Active logics: A unified formal approach to episodic reasoning. Technical report, Department of Computer Science, University of Maryland, 1999.

[12] D. Gabbay. *Labelled Deductive Systems, Vol. 1*. Oxford University Press, 1996.

[13] T. O. Heins. A case study of active logic. Master's thesis, Department of Computer Science, Lund University, January 2009. Available at `http://fileadmin.cs.lth.se/ai/xj/-ThorbenHeins/report.pdf`.

[14] F. Heintz. *DyKnow. A Stream-Based Knowledge Processing Middleware Framework*. PhD thesis, Department of Computer Science, Linköping University, Sweden, 2009. Linköping Studies in Science and Technology, Dissertation No. 1240.

[15] C. Hewitt. Common sense for concurrency and strong paraconsistency using unstratified inference and reflection. Technical report, `http://carlhewitt.info`, 2008. Available at: `http://commonsense.carlhewitt.info`.

[16] M. Nirkhe, S. Kraus, and D. Perlis. Situated reasoning within tight deadlines and realistic space and computation bounds. In *Proc. Common Sense 93*, 1993.

[17] K. Purang, D. Purushothaman, D. Traum, C. Andersen, D. Traum, and D. Perlis. Practical reasoning and plan execution with active logic. In *Proceedings of the IJCAI'99 Workshop on Practical Reasoning and Rationality*, 1999.

[18] B. Selman and H. Kautz. Knowledge compilation and theory approximation. *JACM*, 43(2):193–224, 1996.

Classifying the Severity of an Acute Coronary Syndrome by Mining Patient Data

Niklas Lavesson* Anders Halling† Michael Freitag‡ Jacob Odeberg§
Håkan Odeberg¶ Paul Davidsson‖

Abstract

An Acute Coronary Syndrome (ACS) is a set of clinical signs and symptoms, interpreted as the result of cardiac ischemia, or abruptly decreased blood flow to the heart muscle. The subtypes of ACS include Unstable Angina (UA) and Myocardial Infarction (MI). Acute MI is the single most common cause of death for both men and women in the developed world. Several data mining studies have analyzed different types of patient data in order to generate models that are able to predict the severity of an ACS. Such models could be used as a basis for choosing an appropriate form of treatment. In most cases, the data is based on electrocardiograms (ECGs). In this preliminary study, we analyze a unique ACS database, featuring 28 *variables, including: chronic conditions, risk factors, and laboratory results as well as classifications into MI and UA. We evaluate different types of feature selection and apply supervised learning algorithms to a subset of the data. The experimental results are promising, indicating that this type of data could indeed be used to generate accurate models for ACS severity prediction.*

*Blekinge Institute of Technology, Box 520, SE-372 25 Ronneby, email: Niklas.Lavesson@bth.se

†Blekinge Competence Center, Vårdskolevägen 5 SE-371 41 Karlskrona, email: Anders.Halling@ltblekinge.se

‡Herlev Hospital, Region Hovedstaden, Herlev Ringvej 75, 2730 Herlev, Denmark

§Dept. of Medicine, Karolinska Institutet and University Hospital, 171 77 Stockholm

¶Blekinge Competence Center, Vårdskolevägen 5, SE-371 41 Karlskrona, email: hakan.odeberg@ltblekinge.se

‖Blekinge Institute of Technology, Box 520, SE-372 25 Ronneby, email: Paul.Davidsson@bth.se

1 Introduction

The ability to identify patients at high risk of morbidity or mortality grows in importance as a consequence of the increasing ability of modern medicine to provide costly but potentially beneficial treatment [3]. Heart disease is the single most common cause of death for both men and women in the developed world [12]. Moreover, it is also one of leading causes of morbidity and mortality in developing countries such as China [4].

When patients with chest pain arrive at the hospital, the physician needs to make an initial diagnosis. However, the consequences of diagnostic errors can be significant for both patients and their physicians [11]. It would therefore be beneficial if the severity of each case could be determined with greater certainty at this initial stage.

The aim of this preliminary study is to investigate the possibility of automatically generating models (classifiers) that can be used to support the diagnosis of Acute Coronary Syndrome (ACS) patients. ACS Patients are difficult to diagnose and they represent a heterogeneous group with different treatment options. Especially for patients presenting to the hosptial early after debut of symptoms and without characteristic electrocardiogram changes of larger myocardial infarction (ST-elevation, see below), no single laboratory marker/test in clinical use today has sufficient diagnostic specificity and sensitivity. Hence, the diagnosis of ACS patients using a data mining approach would be advantageous in many situations [5].

Based on the chronic conditions, risk factors, and laboratory results of a patient, the generated classifier would suggest a diagnosis for that patient. In addition, some types of classifiers are able to motivate their diagnoses by providing rules or trees

that describe the decision process. Unlike opaque models, these transparent classifiers can be used by physicians and other professionals in order to better understand which factors influenced the diagnosis. The decision rules and trees may also contribute to the generation of hypotheses regarding ACS. The outline for the remainder of this paper is as follows. First, we give a more in-depth description of the problem from a medical point of view. This is followed by a review of related work and a presentation of our approach as well as the aims and objectives of this study. We then describe the data mining experiments and follow up with a review of the results. Finally, we draw conclusions and present some pointers to future work.

1.1 Background

An arteriosclerotic plaque, in the context of the heart, is a swelling in artery walls that contain lipids, calcium and connective tissue. Thrombosis is the formation of a clot or thrombus inside a blood vessel, obstructing the flow of blood through the circulatory system. Thrombosis over plaques occurs because of two different mechanisms, one being endothelial erosion, which could lead to a thrombus being adherent to a plaque. The second mechanism is referred to as plaque disruption, or rupture. Thrombosis is a trigger for cardiac ischemia [13]. An ACS is a set of clinical signs and symptoms, interpreted as the result of cardiac ischemia, or abruptly decreased blood flow to the heart muscle. The subtypes of ACS include Unstable Angina (UA), Non-ST Segment Elevation Myocardial Infarction (NSTEMI), and ST Segment Elevation Myocardial Infarction (STEMI).

The Carlscrona Heart Attack Prognosis Study (CHAPS) [6, 9] has recruited patient material for 843 patients with ACS in Karlskrona during 1992-1996. The material includes 494 patients diagnosed with MI and 349 additional patients diagnosed with UA. For each patient, a number of variables concerning chronic conditions, risk factors, and laboratory results were gathered, including: glucose levels, smoking, hypertension, occurrence of hypercholesterolemia. The laboratory results can be available during the initial evaluation of the patients.Also genetic variables are determined exemplified by the common prothrombotic single polymorphism (Glu298Asp) which affects the function of the endothelial Nitric Oxide Enzyme (eNOS) and thereby availability of NO, an important modulator of hemostasis and vascular tone. There is no distinction between NSTEMI and STEMI cases in the CHAPS database. In other words, both of these subtypes are expressed as type MI. An elevation of the ST-segment of the electrocardiogram indicates a severe transmural ischemia in contrast to the ischemia in NSTEMI which only engage the inner part of the myocardium.

1.2 Related Work

The classification or prediction of coronary heart disease has been extensively studied by the machine learning and data mining communities. For example, the diagnosis of MI was featured as a case study when the CART algorithm was first presented [3]. Additionally, the STATLOG project included a heart disease database, containing 13 attributes, in one of the first large-scale comparative studies on machine learning algorithms [7]. A more recent study [1] uses multivariate regression and recursive partitioning analysis to allow the construction of decision rules and of a neural tree for diagnosis. The performance results, as measured with the area under the ROC curve, are quite good. However, the choice of algorithms and their parameter configurations are not described in detail in the paper, which makes it difficult to perform comparisons. On the contrary, another study properly documents four data mining algorithms and their performance on a data set of more than 1,000 patients but fails to describe the data set attributes [4]. In addition, Artificial Neural Network (ANN) ensembles and Logistic Regression models trained on data from 634 patients have been compared in terms of the Area Under the ROC curve (AUC) [5]. The database consisted of electrocardiograms (ECGs) and data that were immediately available at patient presentation. Results indicate that ANNs outperformed Logistic Regression Models. Several studies have also been conducted on the prognosis of patients. For example, one such study [8] investigated the use of ANNs to predict 30 day adverse outcomes from ACS. The setup of variables as featured in the CHAPS database has not been previously studied in data mining research.

2 Method

In this preliminary study we use a quantitative approach to evaluate the suitability of the CHAPS database as a basis for generating ACS prediction models with data mining algorithms. The CHAPS database has been stratified and divided into two separate sets for training/testing and validation, respectively. In this paper, we will focus on the training/testing set in order to determine which types of algorithms are appropriate for the studied problem. The objectives are to compare the default configurations of commonly applied opaque and transparent data mining algorithms and to perform an initial analysis to determine which factors are relevant for accurate classification of ACS patients. The aim is to gain basic knowledge about model generation from the CHAPS database to enable further and more detailed studies on a smaller number of suitable data mining algorithms.

3 Experiment

The experiment is organized as follows. The CHAPS training/testing data set is first converted to the open source ARFF format to allow for analysis with the Weka machine learning workbench [14]. In order to enable the careful scrutiny and repeatability of evaluation results reported, our description of the results is accompanied with all relevant details. Exact parameter specifications are given when the Weka default parameter configuration has not been used. Table 5 includes the complete list of data set attributes along with descriptions as well as possible values (nominal attributes) or the mean and standard deviation (numeric attributes).

3.1 Data Set Analysis

The training/testing data set consists of 422 instances (subjects) classified as either MI (247 instances) or UA (175 instances). In addition to the class attribute, there are 8 nominal attributes and 19 numeric attributes. The nominal attributes are highlighted in Table 1. For each possible attribute value, we have indicated the number of UA and MI cases along with prior probabilities, p. For each value we also give the odds of MI. The attribute and value pairs with the highest odds are marked with bold. The highest odds for MI classification are given by diabetes = yes followed by eNOS = snphomo and smoking = yes. The numeric attributes have been omitted from this part of the analysis since they need to be discretisized for this purpose.

3.2 Initial Performance Evaluation

We first performed an analysis of the complete set of attributes in the training/testing set (422 instances) by comparing the results of 20 data mining algorithms and a baseline algorithm (ZeroR). We used the Weka default configurations for all algorithms except K-nearest Neighbor (IBk) for which we used $k = 10$ to distinguish it from One-nearest neighbor (IB1). Each algorithm was evaluated by averaging the results of 10 runs of 10-fold cross-validation tests with an initial random seed of 1. We recorded results for two quite different evaluation metrics; accuracy (ACC) and the Area Under the ROC curve (AUC). The results, in terms of both ACC and AUC, are presented in Table 2.

The baseline algorithm, ZeroR, generates classifiers consisting of a single rule with zero antecedents and the majority class as the consequent. Thus, they classify all instances as belonging to the MI class. Since $n = 247$ for the MI class and $n = 175$ for the UA class, ZeroR yields an accuracy score of $247/(247 + 175) = 0.59$. The AUC metric was calculated with respect to UA. Thus, UA instances represent the positive cases and MI instances represent the negative cases. Consequently, the True Positives Rate (TPR) depicts the rate of correct UA classifications and the False Positives Rate (FPR) depicts the rate of MI cases classified as UA. With regard to AUC, the baseline behaves as a random guesser, thus it yields an AUC score of 0.50.

The best AUC score was achieved by the Logistic algorithm (0.74) followed by AdaBoostM1 and Bagging (0.73) while Support Vector Machines (SMO) achieved the best ACC score (0.70) followed by Logistic and Bagging (0.69). When averaging across the two metrics, the overall best performing algorithms were: Logistic and Bagging (0.71), followed by AdaBoostM1 and BayesNet (0.70), and SMO (0.69).

Table 1: Nominal attribute statistics

Attribute	Values	Classification				total	p	MI
		MI	p	UA	p			odds
sex	male	178	0.72	118	0.67	296	0.70	**1.07**
	female	69	0.28	57	0.33	126	0.30	0.86
	missing	0	0.00	0	0.00	0	0.00	
hypertension[a]	no	175	0.71	126	0.72	301	0.71	**0.98**
	yes	64	0.26	47	0.27	111	0.26	0.96
	missing	8	0.03	2	0.01	10	0.02	
diabetes	no	180	0.73	144	0.82	324	0.77	0.89
	yes	52	0.21	20	0.11	72	0.17	**1.84**
	missing	15	0.06	11	0.06	26	0.06	
heart_failure[b]	no	211	0.85	142	0.81	353	0.84	**1.05**
	yes	28	0.11	31	0.18	59	0.14	0.64
	missing	8	0.03	2	0.01	10	0.02	
diabetes_treatment	no	202	0.82	152	0.87	354	0.84	0.94
	pills	7	0.03	0	0.00	7	0.02	0.00
	insulin	4	0.02	4	0.02	8	0.02	0.71
	diet	26	0.11	17	0.10	43	0.10	**1.08**
	missing	8	0.03	2	0.01	10	0.02	
smoking	no	172	0.70	144	0.82	316	0.75	0.85
	yes	63	0.26	28	0.16	91	0.22	**1.59**
	missing	12	0.05	3	0.02	15	0.04	
hypercholesterolemia	no	231	0.94	155	0.89	386	0.91	**1.06**
	yes	8	0.03	18	0.10	26	0.06	0.31
	missing	8	0.03	2	0.01	10	0.02	
eNOS	wildhomo	114	0.46	95	0.54	209	0.50	0.85
	hetero	107	0.43	69	0.39	176	0.42	1.10
	snphomo	26	0.11	11	0.06	37	0.09	**1.67**
	missing	0	0.00	0	0.00	0	0.00	

[a] Treated for high blood pressure
[b] Treated for dysfunction of the heart muscle pump

Table 2: Initial results on the complete set of attributes

Algorithm	Type	AUC	ACC
AdaBoostM1	opaque	0.73(0.07)	0.68(0.07)
Bagging	opaque	0.73(0.07)	0.69(0.06)
BayesNet	opaque	0.72(0.07)	0.67(0.07)
Dagging	opaque	0.69(0.09)	0.65(0.06)
DecisionStump	opaque	0.68(0.06)	0.67(0.07)
HyperPipes	opaque	0.54(0.06)	0.58(0.03)
IB1	opaque	0.54(0.08)	0.54(0.08)
IBk ($k = 10$)	opaque	0.59(0.09)	0.57(0.08)
Logistic	opaque	0.74(0.07)	0.69(0.07)
MLP[a]	opaque	0.65(0.09)	0.62(0.08)
NaiveBayes	opaque	0.69(0.07)	0.58(0.06)
RandomForest	opaque	0.67(0.09)	0.64(0.08)
RBFNetwork	opaque	0.67(0.08)	0.63(0.07)
SMO	opaque	0.68(0.07)	0.70(0.07)
BFTree	transparent	0.67(0.09)	0.68(0.07)
J48	transparent	0.61(0.08)	0.63(0.07)
JRip	transparent	0.65(0.07)	0.66(0.07)
PART	transparent	0.61(0.09)	0.61(0.07)
Ridor	transparent	0.63(0.07)	0.65(0.07)
SimpleCart	transparent	0.69(0.07)	0.67(0.07)
ZeroR	transparent	0.50(0.00)	0.59(0.01)

[a] MultiLayerPerceptron

Table 3: Results on three feature selected data sets

Algorithm	Best First		Nominal		Numeric	
	AUC	ACC	AUC	ACC	AUC	ACC
AdaBoostM1	0.70(0.07)	0.66(0.07)	0.61(0.08)	0.61(0.06)	0.70(0.07)	0.66(0.07)
Bagging	0.75(0.07)	0.62(0.07)	0.62(0.08)	0.61(0.06)	0.68(0.07)	0.58(0.06)
BayesNet	0.76(0.07)	0.71(0.06)	0.61(0.08)	0.60(0.06)	0.75(0.07)	0.70(0.06)
BFTree	0.71(0.08)	0.68(0.07)	0.55(0.09)	0.55(0.06)	0.67(0.08)	0.63(0.06)
Dagging	0.73(0.07)	0.69(0.06)	0.60(0.08)	0.60(0.06)	0.65(0.08)	0.62(0.07)
DecisionStump	0.69(0.06)	0.71(0.06)	0.52(0.04)	0.59(0.05)	0.70(0.06)	0.71(0.06)
HyperPipes	0.72(0.07)	0.70(0.06)	0.57(0.09)	0.56(0.06)	0.66(0.07)	0.64(0.06)
IB1	0.60(0.07)	0.61(0.07)	0.49(0.07)	0.51(0.07)	0.58(0.07)	0.59(0.07)
IBk (k=10)	0.74(0.07)	0.68(0.06)	0.57(0.07)	0.60(0.03)	0.73(0.07)	0.68(0.07)
J48	0.74(0.06)	0.70(0.06)	0.59(0.09)	0.58(0.05)	0.72(0.07)	0.69(0.06)
JRip	0.69(0.08)	0.62(0.05)	0.57(0.10)	0.60(0.04)	0.70(0.08)	0.65(0.06)
Logistic	0.53(0.04)	0.59(0.02)	0.51(0.01)	0.59(0.01)	0.53(0.06)	0.58(0.03)
MLP[a]	0.67(0.08)	0.67(0.07)	0.50(0.06)	0.58(0.04)	0.67(0.09)	0.67(0.07)
NaiveBayes	0.68(0.06)	0.67(0.07)	0.51(0.05)	0.59(0.02)	0.68(0.06)	0.67(0.07)
PART	0.70(0.08)	0.68(0.07)	0.53(0.07)	0.57(0.04)	0.66(0.08)	0.65(0.07)
RandomForest	0.70(0.08)	0.66(0.06)	0.52(0.09)	0.54(0.07)	0.67(0.09)	0.64(0.07)
RBFNetwork	0.69(0.07)	0.68(0.07)	0.51(0.04)	0.57(0.04)	0.69(0.07)	0.67(0.07)
Ridor	0.66(0.07)	0.68(0.07)	0.51(0.06)	0.56(0.06)	0.65(0.08)	0.66(0.07)
SimpleCart	0.68(0.07)	0.65(0.07)	0.57(0.09)	0.57(0.06)	0.68(0.08)	0.65(0.08)
SMO	0.65(0.06)	0.67(0.06)	0.51(0.04)	0.58(0.04)	0.63(0.07)	0.65(0.06)
ZeroR	0.50(0.00)	0.59(0.01)	0.50(0.00)	0.59(0.01)	0.50(0.00)	0.59(0.01)
Average	0.69(0.07)	0.67(0.06)	0.55(0.07)	0.58(0.05)	0.67(0.07)	0.65(0.06)

[a] MultiLayerPerceptron

3.3 Feature Selection

We generated three new data sets using feature selection. The first data set was generated using only numeric attributes (except for the class attribute) while the second data set only features nominal attributes. The third data set, which features 5 numeric and 1 nominal attributes, was generated using the Best First feature selection algorithm [15]. The Best First method is a heuristic search strategy that uses hill climbing and a back-tracking mechanism to reduce the number of attributes and increase the performance [14]. Out of the complete set of attributes, the Best First method selected the following attributes: heart_failure, B_LPK, H1_NEU, B_GLU, B_TMCV, and P_APTT. We again evaluated each algorithm using 10 runs of 10-fold cross-validation tests. The results, which can be viewed in Table 3, indicate that the Best First feature selected data set is the most suitable, since the average AUC and ACC are the highest in comparison to the other data sets, including the data set with the complete set of attributes. BayesNet achieves the highest AUC and ACC, followed by Bagging, J48 and IBk (AUC) and DecisionStump (ACC). Interestingly, Logistic performs poorly on the Best First data set.

3.4 Classifier Understandability

There is often a trade-off between classification performance and understandability. In our experiment, we evaluated several rule and tree based algorithms that are able to produce classifiers that may provide human-understandable visualizations of the classification process. However, the understandability of tree- and rule-based models depends on the complexity of the trees and rule sets. Other models, e.g., generated by SMO, can also be understood in the sense that it can be determined which attributes are important indicators for a particular class. However, related work often seem to treat neural network and support vector machine models as being opaque. As a result, several studies have presented approaches to generate understandable rules from such models, cf. [2]. We provide some rule-based examples in Table 4 and one decision tree can be viewed in Figure 1.

4 Discussion

We used two different evaluation metrics for this purpose. Firstly, we measured the classification accuracy, i.e., the ratio of correctly classified instances. This metric has been the traditional choice for evaluation and it is very straight-forward to use

Table 4: Rule-based classifiers

Algorithm	Classifier
The following rules were produced using the complete set of attributes	
Jrip	IF (**B_LPK** $\leq$ 7.44) THEN **diagnosis** = ua ELSE **diagnosis** = mi

Ridor	IF (**B_LPK** $>$ 8.175) AND (**B_GLU** $>$ 5.935) AND (**P_PT** $\leq$ 101.5)
	AND (**B_MCV** $>$ 82.5)
	THEN **diagnosis** = mi
	IF (**B_LPK** $>$ 8.175) THEN **diagnosis** = mi
	IF (**B_LPK** $>$ 6.345) AND (**B_TMCV** $\leq$ 8.45) AND (**B_GLU** $>$ 5.45)
	THEN **diagnosis** = mi
	ELSE **diagnosis** = ua

ConjunctiveRule	IF (**B_LPK** $>$ 8.095) THEN **diagnosis** = mi
The following rules and the tree in Figure 1 were produced using the Best first selected attributes	

JRip	IF (**B_LPK** $\leq$ 8.2) AND (**B_TMCV** $\geq$ 9) THEN **diagnosis** = ua
	IF (**B_LPK** $\leq$ 7.41) THEN **diagnosis** = ua
	ELSE **diagnosis** = mi

PART	IF (**B_LPK** $>$ 8.09) AND (**B_GLU** $>$ 5.92) AND (**heart_failure** = no)
	THEN **diagnosis** = mi
	IF (**B_LPK** $>$ 8.81) THEN **diagnosis** = mi
	IF (**B_GLU** $\leq$ 4.6) THEN **diagnosis** = ua
	IF (**B_TMCV** $\leq$ 8.5) AND (**H1_NEU** $>$ 4.18) THEN **diagnosis** = mi
	ELSE **diagnosis** = ua

as well as being easily explainable. However, it suffers from the assumption that the class distribution is known for the target domain and it also assumes equal misclassification costs [10]. These two assumptions are rarely met in real-world problems and the studied problem is a perfect example of this. Thus, we also calculated the area under the ROC (AUC) metric for the purpose of classifier evaluation. This metric does not suffer from the two earlier mentioned assumptions; however, it does suffer from an information loss in comparison to a complete ROC plot. To summarize, there are known issues with most of the currently used evaluation metrics, but we argue that the combined information gained from the ACC and AUC evaluations is adequate for the purpose of this preliminary study. We first generated classifiers using the complete set of attributes. The best performing classifiers achieved an accuracy score of 0.70 and an AUC score of 0.74, while the worst performing classifiers behaved like random guessers. These results may be satisfactory for real-world diagnosis purposes; however, we assumed that the results could be improved by reducing the dimensionality of the data set in terms of the number of input attributes. We therefore proceeded by applying a feature selection algorithm to reduce the number of attributes. We used the Best First feature selection method and succeeded in reducing the number of attributes from 27 to 6 while increasing both ACC and AUC for most algorithms. However, the increase in performance was only slight. The best performing classifier now achieved an accuracy score of 0.71 and an AUC score of 0.76.

5 Conclusions

This preliminary study has investigated the potential for using data mining methods to find useful patterns in an Acute Coronary Syndrome (ACS) patient data set. If found, such patterns could be used to generate classifiers that would aid the diagnosis of future ACS subjects. We have trained and evaluated 20 well-known data mining algorithms on different variations of a set of 422 instances. Each instance describes a patient by using 27 input attributes, diagnosed as either Unstable Angina ($n = 175$) or Myocardial Infarction ($n = 247$). The performance results are promising; however, we speculate that the access to more training data and careful parameter tuning could increase the performance further. This study also shows that the featured opaque classifiers perform better than the transparent (understandable) classifiers. This makes it interesting to further explore the trade-off between classification performance and understandability. However, one notable exception to this rule is the J48 tree inducer, which managed to achieve an AUC score of 0.74 on the Best First

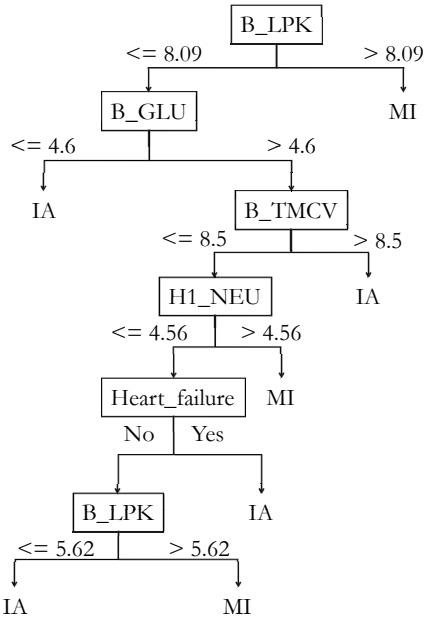


Figure 1: J48 decision tree with 6 branches and 7 leaves

data set. Perhaps most interestingly, most learning algorithms, as well as the feature selection algorithm, tended to agree on the importance of at least two attributes: B_LPK and B_GLU. For example, JRip managed to achieve an accuracy of 0.66 by generating a rule based only on B_LPK. There are a number of interesting directions for future work. Firstly, we would like to establish which feature selection method is the most suitable for the domain. We also intend to perform extensive algorithm parameter tuning in order to generate better models by concentrating on the best performing algorithms from this study. The aim is to validate the results of these new models by perform evaluations on the previously unseen validation data set. Thirdly, we will perform a deeper analysis of the featured attributes and investigate correlations between them. We would also like to introduce additional attributes describing inflammatory markers that may be suitable indicators of the severity of an ACS outcome.

Acknowledgments

This work was partly funded by Blekinge County Council.

References

[1] R. Bassan, L. Pimenta, M. Scofano, and J. F. Soares. Accuracy of a neural diagnostic tree for the identification of acute coronary syndrome in patients with chest pain and no st-segment elevation. *Critical Pathways in Cardiology*, 3(2):72–78, 2004.

[2] Ricardo Blanco-Vega, José Hernández-Orallo, and M. José Ramírez-Quintana. Analysing the trade-off between comprehensibility and accuracy in mimetic models. In *Discovery Science*, pages 338–346, 2004.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. 1984.

[4] J. Chen, Y. Xing, G. Xi, J. Chen, J. Yi, D. Zhao, and J. Wang. A comparison of four data mining models: Bayes, neural network, svm and decision trees in identifying syndromes in coronary heart disease. In *Fourth International Symposium on Neural Networks*, 2007.

[5] M. Green, J. Björk, J. Forberg, U. Ekelund, L. Edenbrandt, and M. Ohlsson. Comparison between neural networks and multiple logistic regression to predict acute coronary syndrome in the emergency room. *Artificial Intelligence in Medicine*, 38(3):305–318, 2006.

[6] K. Holmberg, M-L. Persson, M. Uhlén, and J. Odeberg. Pyrosequencing analysis of thrombosis-associated risk markers. *Clinical Chemistry*, 51:1549–1552, 2005.

[7] R. D. King, C. Feng, and A. Sutherland. STATLOG: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, 9(3):259–287, 1995.

[8] C. L. McCullough, A. J. Novobilski, and F. M. Fesmire. Use of neural networks to predict adverse outcomes from acute coronary syndrome

for male and female patients. In *Sixth International Conference on Machine Learning and Applications*, 2004.

[9] J. Odeberg, M. Freitag, H. Odeberg, L. Råstam, and U. Lindblad. Severity of acute coronary syndrome is predicted by interactions between fibrinogen concentrations and polymorphisms in the GPIIIa and FXIII genes. *Thrombosis and Haemostasis*, 4:909–912, 2006.

[10] Foster Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *15th International Conference on Machine Learning*, pages 445–453, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers.

[11] R. A. Rusnak, T. O. Stair, K. Hansen, and J. S. Fastow. Litigation against the emergency physician: Common features in cases of missed myocardial infarction. *Annals of Emergency Medicine*, 18:1029–1034, 1989.

[12] H. Tunstall-Pedoe, K. Kuulasmaa, P. Amouyel, D. Arveiler, A. M. Rajakangas, and A. Pajak. Myocardial infarction and coronary deaths in the World Health Organization MONICA project. registration procedures, event rates, and case-fatality rates in 38 populations from 21 countries in four continents. *Circulation*, 90(1):583–612, 1994.

[13] L. Wallentin and B. Lindahl A. Siegbahn. Unstable coronary artery disease. In E. Falk, editor, *Textbook of Cardiac Disease*. Mosby, New York, 2002.

[14] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2005.

[15] L. Xu, P. Yan, and T. Chang. Best first strategy for feature selection. In *Ninth International Conference on Pattern Recognition*, pages 706–708, New york City, NY, USA, 1988. IEEE Press.

Table 5: Data Set Description

Attribute	Values[a]	Description[b]
sex	male,female	
age	63.8(8.78)	
hypertension	no,yes	
diabetes	no,yes	
heart_failure	no,yes	
diabetes_treatment	no,pills,insulin, diet	
smoking	no,yes	
hypercholesterolemia	no,yes	
eNOS	wildhomo[c],hetero,[d] snphomo[e]	endothelial Nitric Oxide Synthase
B_LPK	8.94(3.03)	B-Leucocytes
B_HB	136.7(14.5)	B-Hemoglobin
B_EVF	40.5(4.24)	B-Hematocrit
B_MCV	90.37(5.43)	B-Erytrocyte Mean Corpuscular Volume
B_TROM	226.1(63.7)	B-Thrombocytes
H1_NEU	6.27(2.78)	B-Neutrophils
P_PT	83.97(24.49)	P-Prothrombin Time
S_KREA	101.9(71.43)	S-Creatinine
S_ALB	38.33(3.65)	S-Albumin
S_KOL	6.17(1.32)	S-Cholesterol
S_HDLKOL	1.17(0.38)	S-HDL-Cholesterol
B_GLU	6.78(3.05)	B-Glucose
S_TSH	2.18(3.04)	S-Thyroid-Stimulating Hormone
B_TMCV	9.04(0.75)	B-Thrombocyte Mean Corpuscular Volume
P_APTT	33.17(21.60)	P-Activated Partial Thromboplastin Time
S_TG	2.03(1.43)	S-Triglycerides
S_HBA1C	5.24(1.31)	S-Hemoglobin A_{1C}
P_FGEN	3.69(0.94)	P-Fibrinogen
diagnosis	mi,ua	

[a] Given as the complete set of categories (nominal) or the mean and SD (numeric)
[b] The laboratory samples are of type: Blood (B), Serum (S), or Plasma (P)
[c] Wild-type homogeneous eNOS
[d] Heterogeneous eNOS
[e] Single-nucleotide polymorphism eNOS

An Overview on Recent Medical Case-Based Reasoning Systems

Shahina Begum, Mobyen Uddin Ahmed, Peter Funk, Ning Xiong

School of Innovation, Design and Engineering, Mälardalen University,
PO Box 883 SE-721 23, Västerås, Sweden
Tel: +46 21 10 14 53, Fax: +46 21 10 14 60
{firstname.lastname}@mdh.se

Abstract

Case-based reasoning systems for medical application are increasingly applied to meet the challenges from the medical domain. This paper looks at the state of the art in case-based reasoning and some systems are classified in this respect. A survey is performed based on the recent publications and research projects in CBR in medicine. Also, the survey is based on e-mail questionnaire to the authors' to complete the missing property information. Some clear trends in recent projects/systems have been identified such as most of the systems are multi-modal, using a variety of different methods and techniques to serve multipurpose i.e. address more than one task.