Toward Semantic Interoperability for Software Systems

Kendall Lister

Submitted in total fulfilment of the requirements of the degree of Doctor of Philosophy

May 2008

Department of Computer Science and Software Engineering
The University of Melbourne
TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS
Abstract

“In an ill-structured domain you cannot, by definition, have a pre-compiled schema in your mind for every circumstance and context you may find ... you must be able to flexibly select and arrange knowledge sources to most efficaciously pursue the needs of a given situation.” [57]

In order to interact and collaborate effectively, agents, whether human or software, must be able to communicate through common understandings and compatible conceptualisations. Ontological differences that occur either from pre-existing assumptions or as side-effects of the process of specification are a fundamental obstacle that must be overcome before communication can occur. Similarly, the integration of information from heterogeneous sources is an unsolved problem. Efforts have been made to assist integration, through both methods and mechanisms, but automated integration remains an unachieved goal. Communication and information integration are problems of meaning and interaction, or semantic interoperability. This thesis contributes to the study of semantic interoperability by identifying, developing and evaluating three approaches to the integration of information. These approaches have in common that they are lightweight in nature, pragmatic in philosophy and general in application.

The first work presented is an effort to integrate a massive, formal ontology and knowledge-base with semi-structured, informal heterogeneous information sources via a heuristic-driven, adaptable information agent. The goal of the work was to demonstrate a process by which task-specific knowledge can be identified and incorporated into the massive knowledge-base in such a way that it can be generally re-used. The practical outcome of this effort was a framework that illustrates a feasible approach to providing the massive knowledge-base with an ontologically-sound mechanism for automatically generating task-specific information agents to dynamically retrieve information from semi-structured information sources without requiring machine-readable meta-data.

The second work presented is based on reviving a previously published and neglected algorithm for inferring semantic correspondences between fields of tables from
heterogeneous information sources. An adapted form of the algorithm is presented and evaluated on relatively simple and consistent data collected from web services in order to verify the original results, and then on poorly-structured and messy data collected from web sites in order to explore the limits of the algorithm. The results are presented via standard measures and are accompanied by detailed discussions on the nature of the data encountered and an analysis of the strengths and weaknesses of the algorithm and the ways in which it complements other approaches that have been proposed.

Acknowledging the cost and difficulty of integrating semantically incompatible software systems and information sources, the third work presented is a proposal and a working prototype for a web site to facilitate the resolving of semantic incompatibilities between software systems prior to deployment, based on the commonly-accepted software engineering principle that the cost of correcting faults increases exponentially as projects progress from phase to phase, with post-deployment corrections being significantly more costly than those performed earlier in a project’s life. The barriers to collaboration in software development are identified and steps taken to overcome them. The system presented draws on the recent collaborative successes of social and collaborative on-line projects such as SourceForge, Del.icio.us, digg and Wikipedia and a variety of techniques for ontology reconciliation to provide an environment in which data definitions can be shared, browsed and compared, with recommendations automatically presented to encourage developers to adopt data definitions compatible with previously developed systems.

In addition to the experimental works presented, this thesis contributes reflections on the origins of semantic incompatibility with a particular focus on interaction between software systems, and between software systems and their users, as well as detailed analysis of the existing body of research into methods and techniques for overcoming these problems.
Declaration

I hereby certify that:

(i) this thesis comprises only my original work towards the PhD except where indicated,

(ii) due acknowledgement has been made in the text to all other material used,

(iii) this thesis is less than 100,000 words in length, exclusive of table, maps, bibliographies, appendices and footnotes.
Publications

During the course of this project, a number of public presentations have been made which are based on the work presented in this thesis. They are listed here for reference.


TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS


- Kendall Lister & Leon Sterling, **Reconciling Heterogeneous Information Sources**, in *Proceedings of 3rd International Semantic Web Conference (ISWC ’04)*, Hiroshima, Japan, 2004

Acknowledgements

My supervisor, Professor Leon Sterling, for allowing me the opportunities and space to find my way through the journey that has led to this thesis (and many other valuable experiences), all the while stepping in when necessary to bring me back on to the path to completion.

My co-supervisor, Dr Catherine Legg, for providing rigorous criticisms and valuable suggestions, as well as generous support and inspiration.

Professor Liz Sonenberg for helping me to see connections and structure that otherwise would have remained hidden.

Dr Doug Lenat and the many people at Cycorp for their patience, advice and enthusiasm; thank you Robert, Pace, Cynthia, Blake, Michael, Chip, Jim, Kevin and anyone whose cake I ate.

Dr Thomas Juan for many conversations and much advice, and the occasional cup of sake.

Dr Kuldar Taveter and Juny Kesumadewi for their collaboration and discussion of ideas, code, results and feedback.

Professor Justin Zobel and William Webber for their critical recommendations for experimental evaluation.

Chirag Pathak for collaborating in the purchase and consumption of approximately $10,000 worth of coffee.

My software engineering students for providing welcome distractions from the preparation of this document.

My parents for their unwavering support.

My partner Pei Pei for understanding and putting up with the highs and lows of the past four years, and consistently encouraging me to hurry up and finish, now!
TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS
# Contents

1 Introduction 1  
1.1 The interoperation of software systems 1  
1.2 Programs as models, data as symbols 2  
1.3 Homogeneous and heterogeneous systems 5  
1.4 Motivating thoughts 7  
1.5 Research question and contributions of this thesis 8  

2 Software Agents, Knowledge, Tasks and Interaction 13  
2.1 Distributed knowledge 13  
2.2 Organisational culture affects communication 15  
2.3 Our software colleagues 17  
2.4 Automating conceptualisation 18  
2.5 Simplicity, success and a semantic web 23  
2.6 Local and implicit ontologies 24  

3 Historical Perspective 27  
3.1 Approaches to information integration 27  
3.2 Semantic mark-up languages 32  
3.3 The Semantic Web 34  
3.4 Ontology alignment 36  
3.5 Assisted and automatic ontology alignment 39  
3.6 Evaluating ontology alignment methods 43  

4 Tasks, Context, Cyc and Sports Reports 49  
4.1 Introducing Cyc 54  
4.2 Context in Cyc 55  
4.3 The task at hand 59  
4.4 SportsReporter 62
4.5 Meaning and relevance ........................................ 63
4.6 Building a lightweight information agent ...................... 65
4.7 Integrating SportsReporter into the Cyc ontology .............. 68
4.8 Moving task-defining heuristic knowledge into Cyc .......... 72
4.9 Generalising the SportsReporter heuristics ................... 76
4.10 Conclusions .................................................. 78

5 Automatic Reconciliation of XML Structures ................. 83
5.1 The EBFM Algorithm ........................................ 83
5.2 Extending the EBFM algorithm ................................ 87
5.3 On the similarity of data instances ............................ 92
5.4 Comparing strings and data without domain knowledge ...... 95

6 Applying AReXS to Real, Messy Information ................. 97
6.1 Experimental design ......................................... 98
6.2 Weather reports ............................................ 100
6.3 Stock quotes ................................................ 104
6.4 Second-hand car classified advertisements .................. 105
6.5 On the correctness of semantic matches ..................... 115
6.6 Surveying human judgements ................................ 123
6.7 Results ..................................................... 132
6.8 Discussion .................................................. 138

7 Data Design Collaboration ..................................... 143
7.1 Barriers to aligning data definitions during development .... 144
7.2 Ontology repositories and the Semantic Web ................. 151
7.3 A prototype ............................................... 153
7.4 Evaluation and conclusions .................................. 154

8 Conclusion .................................................... 165
8.1 Experimental results ........................................ 166
8.2 Future work ................................................ 168
# List of Tables

6.1 Hand-chosen field matches for weather reporting web services ........................................ 102
6.2 AReXS field matches for weather reporting web services ............................................. 102
6.3 Precision and recall of AReXS field matches for weather reporting web services ......................... 102
6.4 Precision and recall of AReXS field matches for weather reporting web services considering only strongest match per field .................................................. 102
6.5 Precision and recall of EBFM with five items represented in both information sources [56] ......................... 103
6.6 Precision and recall of EBFM with varying number of items represented in both information sources [56] ............................................................... 103
6.7 Hand-chosen field matches for stock price reporting web services ................................... 106
6.8 AReXS field matches for stock price reporting web services ........................................ 106
6.9 AReXS field matches for stock price reporting web services filtered for best match per field ................................................................. 107
6.10 AReXS field matches for stock price reporting web services filtered for best match per field and threshold of 0.7 ...................................................... 107
6.11 Precision and recall of AReXS field matches for stock price reporting web services considering only strongest match per field with threshold of 0.7 ...................................................... 107
6.12 Survey results for matching fields between second-hand car web sites ................................. 128
6.13 Correct field matches as chosen by the survey respondents after filtering ............................. 131
6.14 My choices for correct field matches between second-hand car web sites after filtering .................. 134
6.15 AReXS filtered field matches for second-hand car sales web sites ........................................ 137
6.16 Precision and recall of AReXS field matches for second-hand car sources measured against survey responses ...................................................... 138
6.17 Precision and recall of AReXS field matches for second-hand car sources measured against my choices  

139
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Main campus faculty definition by Whang, <em>et alia</em></td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>Branch campus faculty definition by Whang, <em>et alia</em></td>
<td>30</td>
</tr>
<tr>
<td>3.3</td>
<td>Global faculty definition by Whang, <em>et alia</em></td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Entrants in the OAEI ontology alignment contest in 2007 [35]</td>
<td>45</td>
</tr>
<tr>
<td>4.1</td>
<td>An extract from the ACM Computing Classification System hierarchy of topics [6]</td>
<td>60</td>
</tr>
<tr>
<td>4.2</td>
<td>Examples of knowledge units used by the <em>SportsFinder</em> information retrieval agent</td>
<td>62</td>
</tr>
<tr>
<td>4.3</td>
<td>Match results for the F.A. Premier League as published on the F.A. Premier League web site</td>
<td>66</td>
</tr>
<tr>
<td>4.4</td>
<td>Example heuristics used in the <em>SportsReporter</em> information retrieval agent</td>
<td>66</td>
</tr>
<tr>
<td>4.5</td>
<td>Predicate <em>winnerInConflict</em> as defined in Cyc</td>
<td>69</td>
</tr>
<tr>
<td>4.6</td>
<td><em>SoccerTeam</em> concept as defined in Cyc</td>
<td>70</td>
</tr>
<tr>
<td>4.7</td>
<td>Constant <em>SoccerTeam</em> in Cyc hierarchy of concepts</td>
<td>71</td>
</tr>
<tr>
<td>4.8</td>
<td>Match results for Italian Serie A League as published on the Soccerstats web site</td>
<td>80</td>
</tr>
<tr>
<td>6.1</td>
<td>Definition of precision and recall metrics</td>
<td>99</td>
</tr>
<tr>
<td>6.2</td>
<td>Definition of precision and recall as applied to AReXS output</td>
<td>99</td>
</tr>
<tr>
<td>6.3</td>
<td>Example record collected from <em>World Weather</em></td>
<td>101</td>
</tr>
<tr>
<td>6.4</td>
<td>Example record collected from <em>Airport Weather</em></td>
<td>101</td>
</tr>
<tr>
<td>6.5</td>
<td>Example record collected from <em>Delayed Stock Quotes</em></td>
<td>104</td>
</tr>
<tr>
<td>6.6</td>
<td>Example record collected from <em>Xignite Quotes</em></td>
<td>105</td>
</tr>
<tr>
<td>6.7</td>
<td>Example record from <em>Autotrader</em></td>
<td>108</td>
</tr>
<tr>
<td>6.8</td>
<td>Search interface for the Autotrader web site</td>
<td>110</td>
</tr>
<tr>
<td>6.9</td>
<td>Car descriptions presented by the Autotrader web site</td>
<td>111</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>6.10</td>
<td>Car descriptions presented by the Autoweb web site</td>
<td>112</td>
</tr>
<tr>
<td>6.11</td>
<td>Car descriptions presented by the Carsales web site</td>
<td>113</td>
</tr>
<tr>
<td>6.12</td>
<td>Car descriptions presented by the Ebay web site</td>
<td>114</td>
</tr>
<tr>
<td>6.13</td>
<td>Example records from the Autotrader source</td>
<td>118</td>
</tr>
<tr>
<td>6.14</td>
<td>Example records from the AutoWeb source</td>
<td>119</td>
</tr>
<tr>
<td>6.15</td>
<td>Example records from the CarSales source</td>
<td>120</td>
</tr>
<tr>
<td>6.16</td>
<td>Example records from the Drive source</td>
<td>121</td>
</tr>
<tr>
<td>6.17</td>
<td>Example records from the Ebay source</td>
<td>122</td>
</tr>
<tr>
<td>7.1</td>
<td>Web site of Yahoo7 showing ‘Most Viewed Stories’</td>
<td>147</td>
</tr>
<tr>
<td>7.2</td>
<td>Web site of the BBC showing ‘See what’s popular and new’</td>
<td>148</td>
</tr>
<tr>
<td>7.3</td>
<td>Web site of The Age showing ‘Today’s Top 10 Articles’</td>
<td>149</td>
</tr>
<tr>
<td>7.4</td>
<td>My projects list in Data Design Collaboration web site</td>
<td>155</td>
</tr>
<tr>
<td>7.5</td>
<td>All projects list in Data Design Collaboration web site</td>
<td>156</td>
</tr>
<tr>
<td>7.6</td>
<td>Project description in Data Design Collaboration web site</td>
<td>157</td>
</tr>
<tr>
<td>7.7</td>
<td>Class diagram for project in Data Design Collaboration web site</td>
<td>158</td>
</tr>
<tr>
<td>7.8</td>
<td>Similar classes list in Data Design Collaboration web site</td>
<td>159</td>
</tr>
<tr>
<td>7.9</td>
<td>Comparing two similar classes in Data Design Collaboration web site</td>
<td>160</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 The interoperation of software systems

Software systems cannot naturally interoperate. That is, unless specifically designed to do so, data produced by one system typically cannot be processed by another, nor can one system call upon the functions of another, without human intervention or planning. This comes from the nature of software: it is, to appropriate Gruber’s oft-quoted definition of an ontology [47], an explicit representation of a conceptualisation. Software begins with ideas, thoughts about what might be done and to what it might be done. After a number of activities, generally referred to as software development, a symbolic model is constructed that captures, to a certain extent, the original ideas. This symbolic model is expressed in a programming language and called source code, which is then either interpreted by another computer program, or compiled by another computer program to produce a new symbolic model. In either case, some physical machinery is then electrically manipulated to cause some effects that, again to a certain extent, mimic the original ideas. The key concept here is the transition from ideas, which are abstract, to a symbolic model, which is concrete, and the imperfection of this transition.

The connection between a relatively abstract idea and a concrete symbolic model of that idea is fundamentally and critically a one-to-many relationship; for any given idea there are many symbolic models that could represent it. In other words, any particular function or operation can be implemented by many different computer programs. A software developer makes many choices when designing a software system to represent an idea: she chooses which concepts to abstract and which to represent in detail, she chooses which data types to use and in what structure. Each software developer modelling a particular idea has many options available for achieving the same end, and
individual factors such as experience, training, style and personality mean that different software developers will choose different options. So not only is it theoretically possible to model a concept or function in different ways, and to implement an operation by different computer programs, but in practice individual human factors ensure that this will happen. Garlan, Allen and Ockerbloom coined the term *architectural mismatch* to identify the problems that they perceive complicate the goal of component re-use in software system development [42], and their taxonomy of assumptions that lead to these problems enumerates the many points at which designers and developers might diverge as they model an idea.

These divergences by themselves, although they complicate the construction of software systems, don’t necessarily impact on the ability of software systems to interoperate, because by their nature as mathematical functions, computer programs and software systems can be considered to be ‘black boxes’, functions which, although their inner workings may be unknown, can be relied upon to produce certain output when given certain input. Because of this, the possibility of an idea to be represented by a multitude of computer programs can be discounted and any computer programs that are ‘functionally equivalent’, that is, for certain input they all produce the same output, can be considered interchangeable.

### 1.2 Programs as models, data as symbols

However, being symbolic models themselves, computer programs can only receive symbols as input and produce symbols as output; these symbols must be elements of a symbolic model and as such are subject to the same weakness as computer programs: any idea can be represented by many different symbolic models. As with the program itself, the choice of model for its input and output lies with the developer, and individual factors will cause different developers to choose different models for the same idea.

Software systems, as symbolic models, can only process symbols from their own model, and cannot correctly process symbols from other models. Thus, in practice, it is common for software systems built to express and perform identical ideas to produce output that, although representing elements from the same idea, are symbols from different models, and thus cannot be correctly processed by the other system. This is the problem of interoperability of software systems.

To reduce this problem it is necessary to reduce the differences between the symbolic models chosen to represent the original ideas. Because computer programs can
be considered as ‘black box’ mechanisms, it is necessary only to reduce the differences of model that relate to those symbols that will be generated as output or expected as input by the programs. A common approach is to take away from individual developers the choice of symbols for these parts of their model; in other words, to define a communication standard to which any computer programs that are required to interoperate must conform (such a standard could be public or private). However, in practice there is a range of definitions of ‘communication’ used when standards are defined, so it is helpful to distinguish different types of communication standards.

A communication protocol specifies a set of symbols that represent information about a communication process, as well as a set of sequences in which these symbols may be acceptably generated and received. In this way, a communication protocol is much like a postal system, being the envelopes and addresses that carry letters and documents from sender to receiver, but saying nothing about their contents. Two computer programs can use a communication protocol to exchange data, but there is no guarantee that the data communicated by one program will be able to be processed by the other. The most well-known, or at least arguably the most widely used, example of a communication protocol today is surely the Hypertext Transfer Protocol (HTTP), used by hundreds of millions of people every day to access billions of web site pages on the Internet.

A communication language, on the other hand, specifies a set of symbols that represent information to be processed by the computer programs generating and receiving the communications. To follow the previous analogy, a communication language is the language in which the letters in the postal system are written. They are meaningful to the sender and receiver, but have nothing to do with the system of envelopes and addresses by which they are being transported and delivered. Simply, data output by a computer program and conforming to a communication language can be processed by any other computer program that can process that communication language. However, the previous example of web site pages on the Internet highlights a complication: web site pages, ostensibly the ‘content’ being transported and delivered by the Hypertext Transfer Protocol, are actually written in (at least) two communication languages: what can be called the meaningful content, written in a natural human language, and instructions about how to format and present the meaningful content, written in a language known as the Hypertext Mark-up Language (HTML). There is thus at least a content language and a presentation language. Further, the Hypertext Mark-up Language specification was recently reformulated as an expression of the Extensible Mark-up Language (XML), adding another layer of linguistic complexity.
Interoperating computer programs highlight the fact that communication consists of many layers. Two canonical examples of this are the OSI's seven layers model [141] and Tim Berners-Lee’s “layer cake” model [13]. The desire to compartmentalise systems into discrete modules or layers is understandable, given the complexities typically involved. Although natural language communication is so complex that it is not fully understood, we manage to interact easily, usually with only infrequent mis-communications. Software systems, on the other hand, require closure, completeness and consistency, so when a problem as complex as interoperation is tackled, progress can only be made by segmenting the problem and solving its components, bounding them and ruling them off once each can be declared to be working correctly. Once a stable, albeit incomplete, foundation has been laid, the reduced complexities that remain unsolved can hopefully be addressed more easily in isolation. Unfortunately, in practice the result is often that the complexities that we do not know how to explicitly represent are merely postponed.

In summary, software systems are symbolic models. For any problem domain, there are many symbolic models that can be created to represent that domain. For a given software application, a subset of its symbolic model is exposed to external entities, via either reception or production. This subset of the symbolic model defines an “interface” between the internal symbolic model of the software application and the outside (virtual) world. These interface sets of symbols are referred to as communication protocols, formats and languages. When designed within the same problem domain, different software systems, as symbolic models, will likely have different interface symbol sets, and therefore will likely be incompatible and thus incapable of meaningful interoperation. This is the problem of semantic interoperability as addressed by this thesis.

I use the term “semantic” to indicate that the considerations of this thesis move beyond the issues involved in connecting computers and software systems with wires and protocols. Beyond interconnection, which is concerned with the transportation of data between applications and systems, lies the problem of enabling software systems to share information, interpreting data in the way that it was intended and in a way that is meaningful to their users. In linguistic terms, “semantic” indicates that attached to each written or spoken symbol or utterance is a meaning, which is signified or indicated by the symbol or utterance, and which is communicated from a writer or speaker to a reader or listener via the symbol or utterance — these symbols and utterances are of course what we commonly call ‘words’. In this sense, then, semantic interoperation is interaction between software systems where the meaning behind the data communicated is preserved.
1.3 Homogeneous and heterogeneous systems

If one imagines systems involving computers to lie on a spectrum of degrees of standardisation of communication, then at one end will be systems in which all aspects of communication are universally defined and all computer programs participating in the system are required to adopt these definitions. Such systems are commonly referred to as homogeneous, suggesting that all constituent entities of the system are alike, at least in terms of those parts of their symbolic models that are used either in communicating to or receiving communications from other entities of the system.

Since we are discussing communication and interaction, there is no need to be concerned with the internal nature of the constituent computer programs, as long as they have all committed to a common model for all communication.

Homogeneous systems are often said to have a shared and global ontology\(^1\), whether explicitly expressed or implied by commitment to a common model, and such an approach can be a successful solution to the problem of semantic interoperability in certain situations. Regardless of the internal nature of each computer program member of such a system, by committing to a common symbolic model for their output they ensure that all members of the system will be able to interact successfully, as a symbol output by a program will have the same meaning to all other programs in the system.

The other end of the ‘standardisation of communication’ spectrum prescribes no a priori commitment to shared or global definitions for symbols; systems of this nature can be called heterogeneous. It is in such systems, for which there is no requirement that symbols be interpreted in the same way by different constituent computer programs, that the problem of semantic interoperability arises.

Although I have just described a spectrum of communication, ranging between completely standardised and completely free, it is really the case that any system in which there is not a commitment to a common model is in fact heterogeneous. However, as I tried to show earlier in this chapter by describing the different levels at which

---

\(^1\)If it seems strange to speak of “an ontology”, as though it is a concrete thing rather than a school of philosophy, there is an explanation: the field of computer science has co-opted the word ontology to name a data structure that encodes the set of concepts and relationships between them that a particular computer program or group of programs are capable of processing. In this sense, “an ontology” is what a more traditional ontologist might produce if asked to document everything that exists in a given world — as mentioned earlier, the most oft-quoted definition is due to Gruber, who wrote that “an ontology is an explicit specification of a conceptualization”[47]. In practice, computer ontologies are typically domain-specific, meaning that they only specify the entities and relationships that exist in a specific area of activity or expertise. In this thesis, I will use ontology in both its traditional sense and to mean the set of concepts known to a software system — I will do my best to make it clear which sense I mean in each instance.
communication can be standardised, it is simpler in practice to consider that systems range from completely homogeneous to increasingly heterogeneous.

From the point of view of addressing the problem of semantic interoperability, homogeneous systems are not very interesting, as they effectively remove the problem by decree. Such systems can still face very difficult engineering challenges, as modelling the concepts and interaction protocols required for a system of computer programs is by no means easy. However, interpreting the meaning of unknown symbols will not be one of those challenges.

A system-wide commitment to a common model introduces restrictions as it solves the problem of semantic interoperability. Once such a commitment has been made, it is no longer possible for new components to be introduced into the system without strictly conforming to the same common model to which all the original components conform. This is problematic for the construction of open systems, for which it might be desirable to allow new software agents to enter and interact, and for the maintenance of systems, for which it might be necessary to introduce new modules and functions. A system-wide commitment to a particular symbolic model makes the system a closed one, and means that any interaction beyond the system’s boundaries that might be desirable after the system’s construction and deployment will face all of the same problems of semantic interoperability that lead to the closing of the system in the first place. In an increasingly connected world, closed systems merely postpone the inevitable.

Heterogeneous systems, on the other hand, do not require global conformity to a particular symbolic model. They can range in formality, from logically complete to *ad hoc*. A logically complete approach may in theory provide complete semantic interoperability, but will inevitably lead to problems of tractability, such as those encountered with the most expressive version of OWL, the Web Ontology Language [46]. While most approaches so far seek to eschew as little completeness as possible in an attempt to retain certainty in the results of implemented systems, lightweight, low-commitment and informal methods can achieve a measure of semantic interoperation, while not guaranteeing results. This thesis describes the outcome of experiments with

---

2The exception to this is when the new component needs only to interact with a subset of the existing components, or via a subset of the defined symbols. Over time, a series of new components could be introduced to a system, each communicating with fewer and fewer of the original components and in increasingly limited ways, while communicating more and more with the other new components. This is easy to imagine, for instance, in the case of a system of legacy components being preserved and slowly retired as a new system is built around them, without a ‘big bang’ change-over. However, I feel that this is a special case that doesn’t detract from the argument that I am making — it is reminiscent of the tale of the woodcutter, who, after replacing his axe’s head and then its handle, wonders whether it is still the same axe.
lightweight approaches to semantic interoperation: scenarios have been identified in which heterogeneous system exists and semantic interoperation is desirable, and tools have been built that can achieve this with a reasonable degree of success.

1.4 Motivating thoughts

Artificial intelligence research has long focused on representing and manipulating knowledge in and by computer programs. The majority of techniques developed for such representation and manipulation have avoided issues of semantic interoperability by centralising the knowledge to be processed by a system or by using the same model for all modules in a system that will process knowledge. Similarly, in the business and public worlds, computer programs have largely been built around centralised knowledge representations. Yet, the ability of computer programs to accept as input data that has been produced as output by another computer program has the potential to greatly extend their usefulness. Early generations of computer programs were typically designed with the assumption that the producers of their input and the consumers of their output would be people. This avoids the issue of semantic interoperability by placing the burden for interpreting and understanding the data on people, who happen to be very good at adapting to new ways of expressing ideas with which they are familiar, but requires that much of the total processing work to be done by the whole system be performed at human speeds, under human availability constraints. Building computer programs that process the data output by other computer programs requires that the programs be designed around the same symbolic model; as described earlier, it is not sufficient that the programs employ the same communication protocol or even language, but their symbolic models must express the intersection of the ideas that they represent in exactly the same way. Failure to do so will result in the programs being unable to meaningfully interact.

For decades, since the creation of structured information stores, repositories and databanks/bases, methods have been employed in order to combine data and information from individual stores. Early efforts involved manually re-entering data from an old store to a new repository; computer programs were then devised to perform these tasks.

The first useful computer program that I wrote as a child was created to achieve this very purpose: to extract data from an obsolete database and convert it into a form that could be imported to a more modern database on a newer operating system and hardware.
Later, as networks began to connect computer systems and software and hardware became more standard and thus remained viable longer, rather than transport data wholesale from an old system to a new one, interfacing programs were constructed to translate data and information from one form to another, allowing new applications access to the contents of older stores and repositories. Such interfacing programs could be referred to as ‘wrappers’, in that they effectively wrapped an old application in a new representation. Following the widespread adoption of the Internet and the World Wide Web as modes of deployment, with the corresponding tendency to provide data repositories and information sources with more user-friendly, graphically-oriented interfaces, the interfacing programs became extracting programs, often referred to as ‘scrapers’ for their practice of virtually rendering web pages into the graphical form comprehensible by people in order to identify and extract the information they require.

Similarly, the next generation of semantically-capable global information infrastructure will necessarily be relatively simple in order to achieve the same scale of acceptance. That is not to say that sophisticated technologies have no place - on the contrary, they will be vital for the areas of industry that require them, and their advances will no doubt drive other research efforts even further. Also, the intelligent agents that roam this infrastructure, and sit on our desktops, will themselves be very sophisticated. However, there remains a fundamental role for simple, flexible and adaptive technologies that do not demand strict adherence to formal standards and protocols and the development and publishing costs that follow. Leaving the majority of the intelligence for semantic comprehension in the interpreting applications rather than the medium itself, can lead to the development of technologies that can operate in any information environment, not just those that are sophisticated and semantically enhanced.

1.5 Research question and contributions of this thesis

This thesis presents an exploration of the problem of semantic interoperability. In Chapter 2, I explore the origins of the problem of semantic interoperability, and consider the most appropriate directions in which to seek solutions. A variety of technologies and approaches have been proposed and applied to different instances of the problems, and in Chapter 3 I survey and assess these. This thesis then presents three experiments in adapting, combining and extending these ideas to solve particular examples of the problem of semantic interoperability, with the goal of advancing the state of the art.

The specific research question that I ask is: given the previous, only partially successful, attempts at automatically reconciling the ontological perspectives of different
software applications and agents, what different approaches can be made and in what situations can they expect to be successful? The three experiments presented in this thesis address this question in different ways. Each experiment is based in real-world situations, wherever possible using real data to demonstrate and evaluate each idea.

The first experiment addresses issues involved in integrating semi- and unstructured information with a large formal ontology, the Cyc knowledge-base. Cyc’s strength is its huge number of common sense rules and facts that have been carefully hand-crafted by professional philosophers and ontological engineers. What Cyc lacks is access to the myriad transient data that are generated every day and are arguably critical for making sensible judgements in the real world, such as daily temperatures, prices, sports results and stockmarket activity, as well as the reams of raw data that would allow the general rules and relationships that Cyc understands to be fleshed out into knowledge useful for interaction with people, such as geographic locations of cities and streets around the world. It is one thing to have defined the logical rules necessary to infer that a monument situated within a city must have longitude and latitude co-ordinates that fall within the physical bounds of that city, but it is quite a different endeavour to manually enter the physical locations of thousands of such monuments and other points of interest around the world.

Yet much of the information just described exists in electronic form already, either within a table of a relational database or on a page of a web site. To import such data en masse into Cyc would be possible, but arguably undesirable for two reasons: firstly, it would likely increase the size of the already very large Cyc knowledge-base by orders of magnitude, and secondly, much of the data of this nature is temporal and needs to be continually updated as new facts are generated as a by-product of human activity. Because much of this data appears either with little semantic decoration (e.g. in a database table) or in a largely unstructured form (e.g. on a web page surrounded by other information), ontological reconciliation is required to transform this data into a form consistent with the concepts and relationships defined within Cyc. The experiment described in Chapter 4 will detail the development a software agent that performs such transformation, and will explore avenues toward the ultimate goal of providing Cyc with the ability to generate such agents as required, allowing it to retrieve and reason about external data as required.

The second experiment deals with the problem of integrating data from multiple information sources, such as databases and catalogue-style web sites, when these sources use different representations for the concepts that they have in common. As is the central theme of this thesis, when people create an explicit representation for concepts,
TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS

they tend to make different choices about the details of that representation. Combined with the ability of software systems only to process symbols in the context in which they were designed, this leads to incompatibilities between software systems and their outputs even though they may be producing data with equivalent meaning. The result in practice has been that in many domains, multiple web sites and other electronic information sources have been created that supply information that users want to compare and analyse, but currently lack the software tools to do so.

Many efforts toward resolving this problem will be presented in Chapter 3, but there is a gap in current research. Most of the approaches surveyed begin by assuming the presence of formal ontologies, and assumes that they can be aligned by analysing their structure and the labels applied to the concepts they contain. However, it is clear that in many cases this level of structure is not available, and the only useful information available is the instances of the concepts in the form of the contents of the data source.

Although two of the previous research works presented in Chapter 3 address situations of this nature, they appear to rely on machine learning techniques, which requires training before being applied to the relevant problem. For the experiment presented in Chapters 5 and 6, I will revive an apparently forgotten algorithm that applies a statistical method to reconcile information sources based on their data instances without training or domain knowledge. In the interests of exploring the extent of the effectiveness of such a technique, I will then apply it to a data set collected from public information source whose contents are highly informal and messy. The algorithm will be compared to the results of a survey of people’s choices of correct alignments.

The third experiment experiment is based on the observation that, as is commonly accepted in the field of software engineering, the later in the development of a software system a fault is discovered and a correction required, the exponentially more it costs. When a semantic incompatibility between two or more software systems is found, it can be regarded as a fault, if the assumption is made that, ideally, all software systems should be capable of interoperation wherever feasible, as is the view, for example, of the Semantic Web project. Given this, I will argue in Chapter 7 that it would therefore be significantly cheaper and easier to correct these incompatibilities before systems are built and deployed, rather than after the fact as is the strategy of most approaches to ontological reconciliation.

There are significant barriers to this, particularly the difficulty of organising communication between developers who may be working for different organisations, in different cities and countries, and the different times and schedules at which systems are generally developed, as well as commercial restrictions on interaction and collaboration.
I will analyse the factors that complicate the resolution of semantic incompatibility prior to development and identify those which I believe can be overcome. Drawing on the increasingly popular model of collaboration provided by open projects such as SourceForge, Wikipedia and Del.icio.us, I will present a prototype web site that enables developers to share data definitions that they have design for use in a software system, and that uses ontology alignment technologies to identify similar data definitions from other projects, thus both alerting developers to the existence of projects that are related to theirs, and providing suggestions for modifications to their own data definitions that would increase the potential for interoperation between systems. In this way, the project that I will propose would attempt to reduce the occurrence of the problem of semantic interoperability via a social, collaborative approach to data definition for information systems, addressing the problem prior to implementation, rather than after deployment.
Chapter 2

Software Agents, Knowledge, Tasks and Interaction

2.1 Distributed knowledge

The issues addressed in this thesis cover a wide range of topics, fields and disciplines. Although it isn’t possible to explore each to the same degree of rigour, I think that it is nonetheless important to introduce the ideas that have motivated, informed and contextualise the works presented here. In this chapter, I want to discuss the background thoughts and observations that have been influential to my work. I will begin with the phenomenon of the enormous quantity of information and knowledge\(^1\) that is being made more and more accessible to people around the world, and then the consideration that knowledge is formed and information created within contexts cultural and historical, and that the norms and conventions of groups, societies and organisations flavour the information that they produce. This flavouring then becomes an obstacle to the communication of information between groups of people, whether they be societies, organisations, teams or even collaborating pairs. I then wish to consider the role that computers and software applications and systems can play in assisting people to locate, comprehend and process information, and the limitations thereof. I will then continue the discussion of homogeneous and heterogeneous systems that I began in the previous chapter, arguing for the application of approaches that incorporate and han-

\(^1\)The meaning of the word ‘knowledge’ is both ambiguous and disputed, and has been for thousands of years. Different schools of philosophical thought provide competing definitions, and lay and casual interpretations also abound. In this thesis I will generally use the term to mean a collection of concepts, relations, assertions and facts. I find particularly useful, when discussing information systems, the notion that information is data with analysis, and knowledge is information with meaning. I am fully aware that each of these terms is rather nebulous, but I beg the indulgence of the reader as I proceed.
TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS

dle heterogeneity, while also identifying and acknowledging the difficulties that come with such approaches. This discussion will cover the typical causes of semantic incompatibility and the role of computer ontologies in addressing these problems. Finally, I will consider the way that these problems manifest on the World Wide Web and the proposed Semantic Web as a motivation for the experiments that will be presented in the subsequent chapters.

That useful knowledge systems inevitably incorporate vast amounts of information is becoming a generally acknowledged phenomenon. The evolution of the computer as a data processing device, and computer networks as communication media, has provided the technical means to aggregate enormous quantities of information. Similarly acknowledged is that our capacity for accumulation, storage and reproduction of data and information has out-paced our ability to perceive and manipulate knowledge. This is not a new realisation: Vannevar Bush identified just such a glut of knowledge and information over fifty years ago and proposed a technological solution in the form of the memex, an enlarged intimate supplement to memory that anticipated the hypertext systems of today [15]. The need for contextualising data today remains thoroughly applicable to the World Wide Web and other large-scale information networks.

By implementing a increasingly global communication infrastructure that provides means for the publication, comparison and aggregation of practically limitless amounts of data, we have discovered the potential to ask questions as individuals conducting our daily lives that previously would have been dismissed as infeasible for anyone less than a dedicated organisation. For example, with the entry cost of publishing a web site effectively negligible, it is difficult to imagine any university not doing so. This means that dozens, or hundreds if we consider a global search, of descriptions of courses, programs and facilities are available for us to peruse. As we learn this, we immediately see a possibility for comparison, and want to ask seemingly simple questions such as “Which faculties offer courses in applied machine vision?” or “Which campuses provide accommodation facilities for post-graduate students?”.

To answer questions like these, we could fairly easily compile a list of university web sites; the list might even be complete. We could then visit each site in turn, browsing or searching and recording what information we think will answer our question. Finally, we could compare the results of our research from each site to formulate an answer. Many people perform this very task every day. The question that interests this thesis is why our computers can’t do this for us yet, and how we can approach the goal of enabling them to do so.
2.2 Organisational culture affects communication

To continue this example, universities as institutions tend naturally to develop and often then actively promote their individuality; this local culture flavours their presentation of information that must then be reconciled with information from other institutions that apply their own cultural characteristics to their publications. If we are to manage knowledge from a variety of sources effectively, we will need the assistance of software that is contextually aware and is capable of negotiating the conflicts that arise when such heterogeneous knowledge is juxtaposed.

To work effectively in the current environment of numerous distributed information systems, knowledge from large numbers of heterogeneous sources must be integrated in such a way that we can efficiently reconcile any differences in representation and context in order to incorporate foreign knowledge into our own world view. To be able to work with knowledge from multiple, often incompatible sources is becoming increasingly necessary [135] as the focus of information processing moves beyond intra-organisational interaction and begins to transgress borders, whether departmental, corporate, disciplinary or ethnic. In 1975, writing on methods for indexing documents for later retrieval, Soergel identified a tension between, on the one hand, the desire to follow as nearly as possible the user’s vocabulary in preparing lists of keywords to represent documents, and on the other hand, to respect as completely as possible the vocabulary of the author and use only terms employed by them to describe a document [118], and argued that there is no such thing as the user or the author. He pointed out that different users, whether as creators or consumers of information, have different viewpoints and naturally use different terminologies, and that an indexer must be able to mediate between these, preserving meaning wherever possible.

Like individual idiosyncrasies, organisational cultures arise as individual organisations develop mechanisms, procedures and representations for dealing with the issues that they face. Because these cultures are generally developed in isolation, each organisation will inevitably arrive at different solutions to what are often very similar problems. In order to stream-line organisational activities and focus group efforts on a common goal, it is necessary for individuals to override their own personal intuited approach to a situation in lieu of an agreed common understanding shared by the other members of the group. We do this naturally when we work together on a problem; some are more able than others, and we recognise teamwork and the ability to understand another’s point of view as desirable qualities. Such qualities are also becoming desirable
TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS

in software systems as software applications and agents\(^2\) term play an increasing role in our communication and collaboration.

When we suppress our own intuitive understanding of a situation and attempt to adopt a standardised, agreed upon approach, we increase our ability to interact with others who have similarly adapted their individual understanding to that of the group or community. But we also lose something in the process: context and generality. An efficient understanding of a situation is like a model, in that the more closely it describes a particular situation, the less effectively it describes a general class of situations. Additionally, as we move from a general conceptualisation of a situation rich with semantic flexibility to a specific understanding, we tend to eschew context. We do this because the very generality that gives us the ability to deal with many varied and new situations is a barrier to communication; at the same time that ambiguity allows adaptation, it prohibits individuals from establishing the certainty of agreement that is necessary for confidence that each understands the other.

However, as organisations discover, standardisation of practices and understandings does not create a panacea for the difficulties of communication and collaboration. On a small scale, adoption of standardised approaches helps individuals to cooperate and achieve goals too large for a single person. On a larger scale, the effort required to establish and prescribe global standards and common approaches grows rapidly beyond feasibility as the number of participants and the amount of data being manipulated increases. As our ability to communicate and interact across cultural borders increases, so does our desire to do so. And as we come to terms with the necessities of increased interoperation and develop coping strategies, if our software tools are to scale similarly

\(^2\)Although there are now whole fields of computing research that consider software ‘agents’ and agent-oriented paradigms, they are of course just computer programs, composed of logical comparisons and data. The term ‘agent’ is typically used to express the idea that a particular class of computer programs is designed to act on behalf of its user, although they are no more capable of acting than any other computer program. The literature often explains that software agents are autonomous, reasoning and even mobile — some researchers speak of software agents moving from one host computer to another — but there is no strict line of definition that can distinguish a ‘regular’ computer program from a software agent. Regardless, there is a definite attractiveness to the concept, as it seems to embody our desires for future software systems, and as such it is a useful metaphor, as long as we retain an awareness that it is just that: a metaphor. With this caution in mind, I will continue to use the term in this thesis, where doing so will save time and effort in communicating the ideas I wish to convey. I will define a software agent as: a computer program that is designed to perform specific tasks on behalf of a user over a period of time, containing within itself the knowledge required to complete the tasks and to interact with other software systems as necessary. By this definition, I would consider a software bot that watches an on-line auction and places bids in the name of a user to be an agent, but I would not apply the term to a program that analyses the log files of a web server and generates charts and statistics. Of course, the nature of computers makes concepts like agency, action, reasoning and decisions vague and problematic, for which reason I do not want to be overly stringent lest this thesis become bogged down in a definitional quagmire.
we must provide them with equivalent reconciliation capabilities.

2.3 Our software colleagues

In many respects, computers are an extreme example of co-workers with poor teamwork and communication skills. When specifying a task for a software application or agent we must specify every step in precise detail, detail that will generally remain constant throughout the life of the software. Whilst people are able to adjust the level of abstraction at which they conceptualise a particular situation, computers traditionally have the capacity only for comparatively very low levels of abstraction. As machines that follow explicit instructions to the letter, their operation is analogous to the most procedural organisational standards, and unsurprisingly they too have great difficulty adapting to new situations.

Traditional computational paradigms require that computer-mediated representations of information and knowledge be exact and literal; for a computer to process information requires simplistic structuring of data and homogeneous representations of concepts. In order to maintain consistency during processing, traditional approaches require that each participant in a system, whether human or software, subscribe to a common understanding of the concepts within the system. In other words, traditional information systems require the adoption of a specified understanding for the domain concerned; deviation from a set of agreed terms and understandings results in a breakdown in communication and loss of consistency through the system.

This ontological homogeneity has worked well for systems with little direct human interaction, when the computers can be left to sort out technical details and people can work at a level removed from the coal face. In fact, isolating technically detailed areas of a system from those areas with which humans interact permits engineering of the technical aspects to create an optimised environment. The World Wide Web is an example of a large-scale system in which the level at which humans interact with the system is greatly separated from the level at which machines interact. We write web pages and read them, navigating along hypertextual paths, while machines manage domain name resolution, protocol selection, transmission of data and rendering of text and images. The gap between the activities of people and machines is highlighted by the problems that occur when we try to make machines work closer to our level as we attempt to automate various functions that we currently perform manually. The example of this most recognisable to the ordinary web user is the task of searching for information, an obviously difficult problem that has yet to be solved to our satisfaction.
But a more far-reaching problem is that of integrating the vast quantities of information available in such a way that we can seamlessly assimilate whatever sources of data are most appropriate to the task at hand, whatever that task may be.

## 2.4 Automating conceptualisation

Automation of data processing is desirable because it frees people from the morass of detail and permits them to utilise their capacity for abstraction. The ability to manipulate concepts at varying levels of detail and to match the level of detail to the needs of the situation at hand is one of our most effective tools for processing knowledge and communicating. Being able to subsume detail within conceptual units of knowledge allows us to overcome the natural limits of our processing capacity; although there appear to be clear cognitive limits on the number of concepts we can articulate at any given time, we have the critical ability to ‘chunk’ collections of knowledge into single units [115, 53], effectively providing a capacity to search through information webs both widely and deeply as necessary. Similarly, when the scope of an information or data problem becomes too great for us to process in a reasonable amount of time, we bring computers to bear on the problem to assist us with storage, recall and simple processing. Automation of data processing will, in principle, provide increased speed and accuracy by reducing the possibility for human error.

By handing low-level information processing tasks to machines, people are freed to consider issues at higher levels of abstraction. If we are to continue to advance the level of assistance that our computers can provide to us as we work, we must elevate our tools to higher levels of abstraction to accommodate the ever-increasing complexity of the situations we face.

As knowledge travels through progressively lower levels of abstraction, its context degrades as generality is replaced by specificity and logical operability. Humans require some specification in order to communicate successfully, but the desired degree of consistency of conceptualisations determines the extent of specification that is necessary. Indeed, it is suggested that even consensus between participants is not always necessary for successful collaboration [11, 120]. As discussed earlier, one of our greatest strengths as humans is our ability to adapt to new situations and reconcile new ontological concepts with our own history of previous experiences. We are also capable of identifying mismatches of understanding in our communications and negotiating shared perspectives as we interact with others [14]. We use the term *ontological reconciliation* for the process of resolving conceptual differences. Human natural language is neither
precise nor predictable, and this seems to reflect the way that we understand the world
though our internal representations and conceptualisations. When we express ourselves
in natural language, we often encounter confusion and difficulty as others attempt to
understand us. This requires us to explore alternative expressions, searching for repre-
sentations that others understand. We do this naturally, and our attention is drawn to
the process only when it fails. But we are generally capable of finding enough common
ground for communication of knowledge to proceed; we are even able to convey basic
information without a common language, as evidenced by the pioneers who encounter
new cultures and learn to translate new languages into their own, and as any tourist
who has managed to gain directions to a restaurant or train station with much waving
of hands can attest.

Fitting knowledge to logical representations is a subjective process. Decisions must
be made about how to express complex concepts in relatively constrained languages;
these decisions are made by people whose choices of representation and expression are
influenced by their own cultural background. Consequently, as context is lost prob-
lems then arise as other organisations with different cultures, or even just individuals
with different conceptualisations, attempt to understand the logical representation and
rebuild the original knowledge. For example, it is widely acknowledged within the
software engineering community that the source code in which computer programs are
written is a notoriously poor medium for communication, and that understanding the
original purpose and reasoning behind a particular piece of source code can be very
difficult.

To return to the case of university web sites, it seems reasonable to assume that
all universities partake in the teaching of students and in research. Most universities
offer undergraduate degrees in the areas of engineering, arts, science and commerce.
But when it comes to describing their activities, where one university may use the
word course to refer to a particular degree program, another will use course to mean
an individual subject within a degree; a third institution may use course to describe a
particular stream or program within a degree. Some institutions will say unit where oth-
ers say subject and others say class. Simply due to their own individual organisational
cultures, different institutions use different vocabularies to describe their activities. The
researcher wishing to compare the services provided by different universities will gen-
erally quickly identify the differences and through an understanding of the knowledge
domain concerning university activities and services will be able to translate between
terms, usually assimilating them into the researcher’s own personal ontological under-
standing, which itself will be shaped by their personal experiences (if they are from
a university that uses *course* to mean a unit of teaching and *program* to describe an undergraduate degree, they will probably translate the descriptions from other institutions into this ontology — if they are not from a particular university, they will probably draw on whatever experience they have of academic institutions, and if they have none, they may build their own ontology from the collection of university representations).

To create software applications and agents that can handle this level of ontological complexity has so far proven to be very difficult. Why then is it preferable to simply agreeing upon a global ontology to which all agents subscribe, a centralised language of understanding and representation, or even a global directory of multiple re-usable ontologies from which agents select as necessary? Ontology creation itself is very difficult, as evidenced by the numerous workshops held every year at academic conferences and by the outcomes or lack thereof from ontology-creation projects such as Cyc [23, 117].

It requires the ability to define many concepts precisely and consistently. It requires the ability to predict appropriate assumptions and generalisations that will be acceptable to most, if not all, people. It also requires universal access and distribution infrastructure, and a well-established and accepted knowledge representation format. It requires some way to address the desire for software applications and agents and humans to interact at variable levels of abstraction as particular situations demand. It requires constant maintenance to ensure freshness and currency, yet also must provide backward compatibility for old agents. It requires that agent developers familiarise themselves with the prescribed knowledge representation formats, ontologies and protocols and adapt their own development efforts to suit them. These issues make a global ontology infrastructure unsuitable as the sole approach, and it is our belief that effort spent adding tolerance of heterogeneity to systems will provide greater benefit as we begin to introduce agents to our multi-cultural world.

In addition to the practical benefits, one of our strongest desires for tolerance of heterogeneity for software systems is rooted unashamedly in idealism: people manage to resolve ontological differences successfully, in real time and ‘on the fly’. This ability gives us much flexibility and adaptability and allows us to specialise and optimise where possible and yet generalise and compromise when necessary. Therefore, it seems both feasible and desirable to have as a goal a similar capability for software agents.

If we are to make effective use of multi-cultural data from heterogeneous sources, we need ways and means to reconcile the differences in representation. If we are to work efficiently to solve large information problems, we need the assistance of automated mechanisms. To achieve both, we need systems that are tolerant of heterogeneity.

Reconciling ontological differences requires understanding the difference between
concepts and their representations; in semiotic terms, appreciating the difference between the signified and the signifier. It has been suggested\(^3\) that while this may be true for a formal ontology and knowledge-base such as Cyc [23], a less formal mechanism such as WordNet [87, 88] provides a counter-example, with its use of synsets: un-labelled groups of synonyms that define a concept or sense. This still, however, implies a distinction between the concept and the symbols that signify it; although the sense or concept is un-named, it is still clearly understood that it is an entity separate from the words that express it.

Reconciling ontological differences means reading multiple texts that represent identical, similar or related concepts and being able to work with them at the concept level rather than at the level of representation. For an XML documents or databases, it might be as simple as realising that two fields in different data sources actually contain the same class of data. On the other hand, it might be as complex as deciding that articles from an economics magazine and an automotive magazine are discussing different topics even though they both have “Ford” and “analysis” in their titles, something that current search technologies based on word frequency and similar techniques would be unlikely to realise.

As the number of data sources available to us and our ability to access them on demand and in real time is increasing, the overhead of pre-constructing a complete ontology for a given interaction becomes less and less viable. Large scale interconnectedness and increased frequency of data transactions across organisational and cultural borders leads to a reduction in the useful life of any context constructed for a particular transaction. Just as we are able to establish contexts and construct suitable local ontologies as needed for particular interactions, if we want to be able to include software agents in our higher level communication and knowledge management, they will need to be capable of similar conceptualisation.

Developing intelligent software agents is a costly and time-consuming exercise. One significant contributing factor to the magnitude of the effort required is the need to conform to prescribed standards for knowledge representation, agent communication languages and ontologies. Such conformity is necessary because without agreement about what may be said and how to say it, communication becomes impossible and the numerous benefits of heterogeneity in multi-agent systems are lost. Two of the most common problems that arise when juxtaposing information from heterogeneous sources are synonymy and polysemy, in which data, labels and markers (in semiotic terms, signs in general) either have the same syntactic representation but have different

\(^3\)In personal communication.
meanings, or have different representations but equivalent meaning [51]. An example of synonymy is that in the context of advertising products for sale, the keywords cost and price will generally be semantically equivalent. As labels, any data to which they refer will tend to be directly comparable. This realisation is necessary for machines to process such data without human intervention, which is the primary goal of current efforts such as the development of a semantic web. The human and cultural factors that lead to this confusion of meaning have been presented earlier in this chapter. The converse of synonymy is polysemy, the phenomenon by which a sign conveys multiple meanings, such as minute, signifying either a unit of time or being very small. Polysemy is perhaps less common than synonymy, but still present in human-created expressions and information, and solutions for reconciliation should deal with both sides of the problem, as do the approaches presented in this thesis.

Formal ontologies are often introduced to represent an explicit statement of the meaning of the data communicated by an information agent or source. However, in a heterogeneous environment even such explicit representations meet many problems when they are combined, as discussed in detail by [60, 75].

To date, global shared ontologies (that is, all participants in a system agree to use a common ontology or design) have been the standard solution to the problem of ontological differences, and they appear to work well in practice for small multi-agent systems [128, 130]. However, their success depends on several characteristics of the agent system for which they are constructed: that the system is relatively small and simple, the ontology does not change, the agents do not change, the life of the system is short, and the context of the system is static. Obviously some of these restrictions are inter-related, and not all will apply to every system, but all place heavy limitations on the long-term, large-scale viability of software agents in open information systems.

A major consequence of the need for an a priori agreed global shared ontology is that the ontology for a system must be designed before the system is implemented. This is necessary because each agent must be equipped with the ontology before they can be initiated into the system. The process of distributing the ontology to all inhabitants of the system is complicated by the size of the system, including the number of agents in it, as well as the complexity of the communication infrastructure within the system. Inevitably, there is a point at which the difficulty of synchronising all agents across a large system to use the same ontology becomes too great. This is not just a technical problem; it is an organisational one as well. The development of the different software agents must be co-ordinated in line with the development of the ontology. As the number of groups or companies involved in the development increases, the admin-
istration required to align their efforts and enforce adherence to the standard ontology becomes overwhelming. A good analogy is the effort of the W3 Consortium to develop the HTML standard. As each new version of the standard was released browsers and servers had to be re-written to cope with the changes; additionally, there was no effective way to enforce compliance when the individual browser developers decided to deviate from the published standard. For a number of years, most web sites that attempted to present anything more sophisticated than plain text with headings were almost certain to render correctly with only one browser, with the result that users were left confused and frustrated. A potential design-time solution to the problem of obtaining agreement for a shared ontology will be introduced later in this thesis.

Predetermined global ontologies also require the prediction of appropriate assumptions and generalisations that will be acceptable to all participants in the system. Likewise, a predetermined ontology tends to inhibit attempts to interact at variable levels of abstraction, as presented earlier in this chapter. Any ontology in a large, long-term system will require maintenance over time, and in an open system backwards compatibility is another requirement to ensure that older agents can continue to operate. These issues all complicate the use of global shared ontologies, and the approaches, techniques and experiments described in this thesis explore possible ways to ease this complication.

2.5 Simplicity, success and a semantic web

The widespread success of the World Wide Web and its underlying technologies, HTML and HTTP, has been due in no small part to their simplicity and ease of adoption. By providing a simple architecture that anyone could learn and use with minimal overhead, content flourished on the web. Other information technologies that arguably provided more effective methods for locating and retrieving data simply failed to take off in the same exponential way that the web did. Where the web infrastructure itself doesn’t even contain the most rudimentary searching and resource location features, Gopher, WAIS and a large number of proprietary on-line databases that predated the World Wide Web all provided automated indexing, searching, hypertextuality and other information management capabilities. But despite their apparent advantages, all of these technologies were overtaken by the web. In fact, in many cases proprietary databases and indexes have had their interfaces replaced with web-based solutions, to the point that the actual technology is largely hidden. It is more than a coincidence that where the World Wide Web succeeded and grew to become a de facto standard, the more
complex alternatives faltered and missed out on popular adoption.

Similarly, I believe that the next generation of semantically-capable global information infrastructure will necessarily be relatively simple in order to achieve the same scale of acceptance. That is not to say that sophisticated technologies have no place — on the contrary, they will be vital for the areas of industry that require them, and their advances will no doubt drive other research efforts even further. Also, the software agents that roam this infrastructure will also be very sophisticated. However, there remains a significant place for simple, flexible and adaptive technologies that do not demand strict adherence to formal standards and protocols and the development and publishing costs that follow. By leaving the majority of the intelligence for semantic comprehension in the interpreting applications rather than the medium itself, we will develop technologies that can operate in any information environment, not just those that are sophisticated and semantically enhanced. There is no suggestion that semantically rich environments are not useful and desirable, but that it is not practical to expect the entirety, or even the majority, of the information landscape of the future to be uniformly structured, as current research seems to imagine.

Discussions of the problems of semantic operability on the web have a tendency to become discussions of the problem of managing and integrating ontologies. The reasons for this are not obscure: ontologies are widely regarded as a critical element of the next generation of data integration solutions, and the World Wide Web is a heterogeneous environment in which foreign data (and therefore ontologies) are regularly juxtaposed. What is less clear is how such data can be combined. A number of new technologies and approaches have been proposed that extend or replace existing web technologies — a selection of these will be presented in Chapter 3. However, the majority of these approaches require either adoption of a specific standard for ontology representation or significant effort to reconcile ontologies. The focus of this thesis is on methods and technologies that seek to minimise the degree of a priori commitment to representation and manual intervention, and it is in this light that the issues presented will be considered. It would, of course, be remiss to discuss semantic interoperability on the web without discussing the Semantic Web project (see for instance [12]) — I will do so in the next chapter.

2.6 Local and implicit ontologies

Most data on the Internet today appears without any explicit ontology. To integrate data that has no accompanying explicit ontological representation requires either that
formal ontologies be constructed for each data source, manually or automatically, or that the conceptual and semantic correspondences between elements in the data be recognised or deduced directly, without resorting to an explicit representation of the ontology. The former process at first seems to be the more reasonable, as it mirrors the intuitive process a person would be likely to define if asked to plan the task (the concept of task-oriented contexts will be developed further later in this thesis). However, the latter process is in fact closer to the actual approach a person would take when given two data sources and asked to reconcile them. Furthermore, the first method introduces several of the most troublesome ontology management issues, namely constructing accurate and usable ontologies, choosing a representation, and then aligning different ontologies. If the two ontologies are developed together, some of the difficulties of the development can be stepped over as the engineer juggles and reconciles concepts and relationships as they go, but such a synergy certainly cannot scale far beyond two data sources at a time. In reality it is often desirable to compare and contrast data from multiple sources, such as a variety of on-line book stores. One objective of our research into ontological reconciliation is to automate the process as much as possible so that any solutions are eventually globally deployable.

Another important benefit of an automated, lightweight approach to ontological reconciliation is that it can make whatever technology is deployed very adaptable to changes in the data environment. If a software agent is tasked with retrieving prices of books from three on-line book stores, traditional ontology development and management approaches require that an engineer assess the data sources and construct mapping ontologies between them. If the companies publishing their stock data does not supply a well-formatted ontology along with the shopping data, the engineer also has to construct three individual ontologies before any mapping can even be considered. If a fourth source of on-line books becomes available, the engineer is required again to construct either another mapping ontology to align the new source and the existing mapping ontologies, or the process must begin again from scratch. Of course, if the software agent has its own ontology for the domain of books sales (which is likely, if it has been designed to search and report on data of this type), it is only necessary for the engineer to construct maps between this ontology and each data source. But each time one of the companies changes their data representation the engineer is again required to manually intervene, unless the company provides sufficient hooks in their ontology for backward compatibility. In the low margin world of on-line commerce, this is hardly likely to be considered a cost-effective effort even though technologies such as SHOE deliberately support this [50]. An automated solution is obviously preferable to one
that requires human supervision, and we suggest that in most end-user applications, the required accuracy is generally not high enough to demand heavyweight tools and processes. Additionally, a well-designed user interface could allow the user to touch up the results of the automatic reconciliation on-the-fly, thus harnessing the intelligence of the user for effectively no cost. The works presented in this thesis will demonstrate promising approaches to the creating of systems that achieve the goals described here.
Chapter 3

Historical Perspective

3.1 Approaches to information integration

Current approaches to semantic interoperability typically assume either that an ontology is available for each information source, containing identifiers for concepts and representations of the relationships between them, or that no ontology is available and nothing more is known about the information sources than their contents. Earlier efforts, on the other hand, tended to be somewhat naive about ontology, treading a middle ground where it was assumed that developers tasked with integrating information sources such as databases would have access to the schema of each database with which they were required to work (and that the labels and names used in the schema would be unambiguous or thoroughly documented) and yet that there would be no explicit specification of the concepts behind the labels or the relationships between them. For the purposes of this thesis I will, perhaps presumptuously, lump together the many efforts to provide logics and frameworks for reasoning about the contents and characteristics of databases and pick several examples to provide a background to the field of semantic interoperability.

Where the earliest methods for integrating databases would consist largely of manual transformations, the combining of logic with database management systems greatly extended what was possible in terms of interoperation between databases. By expressing properties and characteristics of databases themselves, logic-based systems were able to reason about the meanings of the contents of different databases, reducing the amount of manual work required by operators and designers wanting to combine information from multiple database systems. Logics were designed for low-level purposes such as to protect the integrity of databases under uncertain conditions, e.g. unre-
liable transaction environments. Higher-level “meta” logics were developed to handle uncertainty in the contents of the database. In 1994, Subrahmanian described such a logic [127], in which scenarios of the following form are presented and resolved:

Suppose that there are three sensors, leading to databases $DB_1$ (containing radar information), $DB_2$ containing gun characteristics, and $DB_3$ containing information on the speed of a hostile tank. $DB_1$, $DB_2$, and $DB_3$ are given below, together with the observations made by them:

$DB_1$
— Radar reading $r_1$ indicates that the object is a T-72 tank.
— Radar reading $r_2$ indicates that the object is a T-60 tank.
— Radar reading $r_3$ indicates that the object cannot be a T-72 tank.
— Radar reading $r_4$ indicates that the object is a T-80 tank.
— (Observation) The object has radar reading $r_3$.

$DB_2$
— Gun characteristic $c_1$ indicates that the object is a T-60 tank.
— Gun characteristic $c_2$ indicates that the object cannot be a T-80 tank.
— Gun characteristic $c_3$ indicates that the object is a T-72 tank.
— (Observation) The object has gun characteristic $c_3$.

$DB_3$
— High speed indicates that the object cannot be a T-72 tank.
— High speed indicates that the object cannot be a T-80 tank.
— Medium speed indicates that the object cannot be a T-80 tank.
— (Observation) The object has high speed.

Additionally, we know that the object is either a T-72, a T-60, or a T-80 tank.

Note that by using $DB_1$ alone we would conclude that the object is either a T-60 or a T-80 tank. Using $DB_2$ alone, we would conclude that the object is a T-72 (thus contradicting the conclusion yielded by $DB_1$). Finally, $DB_3$ by itself indicates that the object is neither a T-72 nor a T-80 tank (and, hence, presumably a T-60 tank). This contradicts $DB_2$.

Notwithstanding the contradictions present, in the absence of any other data, the autonomous vehicle must draw some conclusion about what the sensed object is. In order to do this, some mechanism for resolving conflicts (fusing sensor information) is necessary. For instance, it may be known that
the gun characteristic sensor is less reliable than the other two. Let us say that this is the only conflict-resolution information that can be reliably used.

- If \( DB_2 \) says “the object is X” has truth value \( v \) and \( DB_1 \) says it has the truth value \( \neg v \), then the conclusion drawn should be that of \( DB_1 \).
- If \( DB_2 \) says “the object is X” has truth value \( v \) and \( DB_3 \) says it has the truth value \( \neg v \), then the conclusion drawn should be that of \( DB_3 \).

Given this conflict-resolution information, it may be concluded that the sensed object is a T-60 tank.

Subrahmanian then presents a full logic that addresses the problem “of integrating multiple deductive databases containing (possibly) inconsistencies, uncertainty, non-monotonic negation, and possibly even temporal information” [127]. The amalgamation of the databases according to this logic is performed by a supervisory database which possesses the necessary “meta-knowledge”, which in the example above would be the statements about the relative reliability of the local databases. Subrahmanian notes a related work that calls these supervisory databases mediators [114], which foreshadows the efforts to develop mediating “middle-agents” later in the 1990s, which in a way bridged the theoretical gap between integrating databases and interoperating information agents.

In 1991, Whang, Navathe and Chakravarthy had proposed a similar logic-based approach to enabling querying multiple heterogeneous databases [136]. They presented the following example of schema integration that their logic could handle:

Consider a university that has a main campus and a branch campus, each of which has its own database of faculties\(^1\). In the database of the main campus, each member of the faculty has a yearly salary and an office; in the database of the branch campus, each member of the faculty has a monthly salary but no office. Possible schema for these databases are given in Figures 3.1 and 3.2. An integrated view of the faculty is given in Figure 3.3. The global schema is derived from the two local schemas: while the attribute name is trivially derived, problems arise for the attributes phone

\(^1\)I am consistently bemused by the semantic interoperability problems created by the different terminology used in universities around the world – while in much of the world “faculty” means a division of a university, usually a collection of academic departments, in America and other places it means the body of teaching staff at an academic institution. In this case Whang et alia mean the latter.
and salary. In particular, for those staff who are members of both faculties, it is assumed that: i) the phone numbers of the main and branch campus are enumerated in list form and ii) the annual salary is computed by adding the salary specified in the main campus database to the annualised salary as computed from the monthly salary specified in the branch campus database.

\[
\text{main\_faculty(name,salary,office#)} \\
\text{office(office#,phone)}
\]

**Figure 3.1:** Main campus faculty definition by Whang, et alia

\[
\text{branch\_faculty(name,salary,phone)}
\]

**Figure 3.2:** Branch campus faculty definition by Whang, et alia

\[
\text{faculty(name,salary,phone,[main\_faculty,branch\_faculty])})
\]

**Figure 3.3:** Global faculty definition by Whang, et alia
The authors then provide logic for querying across both databases, including rules for handling entities that exist in both databases even though the semantics of the databases are not compatible due to their different schemas, such as:

```prolog
faculty(Name,Salary,Phone,ints(main,branch)) :-
    main_faculty(Name,Salary1,Office#),
    office(Office#,Main_phone),
    branch_faculty(Name,Salary2,Branch_phone),
    Phone = [Main_phone, Branch_phone],
    Salary = Salary1 + 12*Salary2.
```

which shows clearly how each faculty member’s telephone numbers are compiled into a list and their salaries are combined correctly to produce a total annual salary. They then go on to show how a general query like:

```sql
SELECT name
FROM faculty.ints(main_faculty,branch_faculty)
WHERE salary > 50,000
```

once transformed into a logical form:

```prolog
query(Name) :-
    faculty(Name,Salary,_,ints(main,branch)),
    Salary > 50,000.
```

can be augmented via the fragment above to give:

```prolog
query(Name) :-
    main_faculty(Name,Salary1,Office#),
    office(Office#,Main_phone),
    branch_faculty(Name,Salary2,Branch_phone),
    Phone = [Main_phone, Branch_phone],
    Salary = Salary1 + 12*Salary2,
    Salary > 50,000.
```

This logic is then deconstructed to identify the sub-queries required to retrieve the relevant records from each of the local databases:
subquery1(Name, Salary) :-
    main_faculty(Name, Salary, _).

subquery2(Name, Salary) :-
    branch_faculty(Name, Salary, _).

which can finally be converted to query expressions in the appropriate language for the local databases:

Main campus using SQL: SELECT name, salary
FROM main_faculty

Branch campus using QUEL: RANGE OF b IS branch_faculty
RETRIEVE (b.name, b.salary)

This example shows how a layer of logic is placed over the databases and query service and, along with some meta-knowledge about the databases, enables simultaneous searching of multiple databases.

In 1998, Taveter introduced the Software Agents for Retrieval of Information (SARI) system [129]. Formalising the ad hoc links and translations employed by the systems discussed previously, SARI used taxonomies expressed in RDF (the Resource Description Framework) to assist in searching across multiple databases, and permitted the manual construction of semantic “bridges” between concepts in the taxonomies associated with the databases to be queried. The difference between SARI and previous efforts to achieve interoperability between databases was the formal description of their contents, which created a new level of abstraction which permits more consistent reasoning about the data contained within and how they could be meaningfully juxtaposed with each other. SARI provided no automation or assistance for the construction of the required bridges, but the application of ontologies to databases was undoubtedly a step in the right direction.

### 3.2 Semantic mark-up languages

The “meta-knowledge” mentioned in the previous section represents a set of known characteristics of databases that can permit decisions to be made about whether their data is comparable, and if so, how best to aggregate their contents. Formalising this “meta-knowledge” facilitated clearer and more consistent reasoning about the data contained within information sources. The Knowledge Query Manipulation Language
(KQML) [40], the Resource Description Framework (RDF) [79] and later the Web Ontology Language (OWL) [76] are the most prominent examples of efforts to provide standard environments for specifying and expressing knowledge about information and information sources in order that computers may process and, in a sense, understand.

Part of the ARPA Knowledge Sharing Effort, KQML was

a message format and a message-handling protocol to support run-time knowledge sharing among agents. KQML can be used as a language for an application program to interact with an intelligent system or for two or more intelligent systems to share knowledge in support of cooperative problem solving.

Designed to address the problems associated with distributed data and heterogeneous processing environments [39], KQML provided a structured, logic-based messaging protocol in which queries and responses between applications, agents and databases can be exchanged in a reasoned manner. It says nothing, however, about the format of the content of messages; as the designers wrote, “[o]ne should be able to use it with any number of content languages ... all the two intelligent agents need to do is agree on a language to use for communication” [39]. It therefore provides a protocol for communication with defined speech acts and expected responses, but leaves aside the issue of how to deal with the information being communicated, which is the main focus of this thesis.

KQML was actually an extension of the Knowledge Interchange Format (KIF) (both were developed through the ARPA Knowledge Sharing Effort), which was itself intended to be a universal “computer-oriented language for the interchange of knowledge among disparate programs” [48]. The KIF project was active in the early 1990s but was set aside in favour of other efforts toward interoperability. In a way, the failure of KIF and KQML to meet the needs of their creators is indicative of the tension between the need to codify machine-processable languages in order to provide confidence in their utility and keeping them flexible in order to avoid limiting their expressivity and breadth of application.

The more recent RDF and OWL are the cornerstone technologies of the Semantic Web project, an effort to augment the current World Wide Web with structures by which “information is given well-defined meanings, better enabling computers and people to work in cooperation” [12]. By marking or tagging words, phrases and data with concepts defined by unique Uniform Resource Identifiers (URIs) whose relationships to other concepts are specified in RDF and OWL, proponents of the Semantic Web
effort claim that “even agents that were not expressly designed to work together can transfer data among themselves when the data come with semantics” [12]. RDF and OWL provide a rich framework within which to express ontologies for the knowledge and information that software agents and applications need to exchange. The problem that then arises is, again, how to resolve the differences that inevitably occur when different people make explicit their conceptualisation of set of concepts.

3.3 The Semantic Web

The Semantic Web project has two stated goals:

- to allow data to be surfaced in the form of real data, so that a program doesn’t have to strip the formatting and pictures and ads off a Web page and guess where the data on it is, and

- to allow people to write (or generate) files which explain — to a machine — the relationship between different sets of data. For example, one is able to make a “semantic link” between a database with a “zip-code” column and a form with a “zip” field that they actually mean the same — they are the same abstract concept. This allows machines to follow links and hence automatically integrate data from many sources [109].

The technologies that will enable these goals to be achieved are, as I mentioned, the RDF and OWL languages, which provide the necessary elements to express the relationships between concepts that will facilitate such reasoning and interoperation. Instead of adopting an arbitrary format to express an ontology or schema for a project, these two tools that have resulted form the efforts of the Semantic Web project have become a standard language for defining data types and relationships between them for semi- and un-structured data, as well as for describing the contents of structured data sources such as databases in terms that enable their integration with other data.

As a political exercise, the Semantic Web project is proceeding very well — it has created a banner under which many researchers, both academic and commercial, can come together to work toward overcoming the semantic gaps and cracks that currently hinder semantic interoperation. However, like KQML, the Semantic Web does not yet address automating ontological reconciliation to assist people in resolving the differences of expression and representation that inevitably arise when specifying information. Instead, semantic tags and labels are simply expressed and then left to be interpreted, compared and aligned by external agents and applications.
CHAPTER 3. HISTORICAL PERSPECTIVE

This is not to say, though, that work is not being done to overcome this, and in a sense the most significant outcome of the Semantic Web project so far has been that many researchers have adapted their ideas and approaches to achieving semantic interoperability to the formats provided by the Semantic Web project. Instead of creating algorithms and tools to assist users to integrate schemas and ontologies expressed in a multitude of formats, it seems that the majority of new approaches assume that the entities to be aligned will be expressed in RDF or OWL, meaning that one more layer has been removed from the interoperability problem.

For example, the OAEI Ontology Alignment Contest [35] described later in this chapter supplies the sample ontologies to be aligned in RDF format, and in fact uses an RDF schema to express the matches produced by each entrant system. In 2004, Heß and Kashmerick presented an application that allows users to annotate web service descriptions expressed in WSDL (the Web Services Description Language) with concepts from an ontology written in OWL [52]. The application presented also applies machine learning to classify the web service descriptions, although it is not clear whether there is any automated assistance for selecting concepts from the ontology to link to elements of the descriptions. Bao, Caragea and Honavar describe a package-based description logic that permits selective importing of concepts from ontologies to enable re-use of ontologies without require commitment to all of their concepts [10], specifically overcoming a weakness of OWL. NASA publishes a set of ontologies expressed in OWL for space-related topics such as sensors, phenomena, space, time, units and substances.

However, giving an indication of the problem that can arise as the number of published ontologies grows, The Marine Metadata Interoperability Project currently publishes 54 ontologies expressed in OWL, all related in some way to marine study. The project also publishes seven settings of mappings between various ontologies, which were prepared during a workshop held in 2005. The project further has created a tool named VINE (Vocabulary Integration Environment) to assist users in creating mappings between ontologies, although it doesn’t appear to be any more sophisticated than the tools described earlier in this chapter, such as PROMPT. There are many other cases of companies and research groups creating domain-specific ontologies using the Semantic Web technologies of RDF and OWL (the Semantic Web Education and Outreach Interest Group currently publish details of seventeen case studies). I intend to discuss some of the efforts related to the Semantic Web in Chapter 7.

---

2 At http://www.planetont.org/
3 According to a page on the project’s web site at http://marinemetadata.org/examples/mmihostedwork/ontologieswork/ontologies/mappings.html/
4 At http://www.w3.org/2001/sw/sweo/public/UseCases/
While discussing the Semantic Web, it is interesting to note that after ten years of effort in building frameworks for ontology-based data marking and linking, some leaders of the project are now looking back to structured databases and applying these tools to opening structured databases to interoperation — “[m]uch of the motivation for the Semantic Web comes from the value locked in relational databases. To release this value, database objects must be exported to the Web as first-class objects and therefore must be mapped into a system of URIs” [110]. This is exactly what the SKSI project, an effort that forms part of the development of Cyc, aims to achieve — this will be discussed in more detail in Chapter 4, where I will present a piece of work that explores the integration of the massive Cyc knowledge-base with an information agent that accesses and translates semi- and structured data sources into a semantically rich form suitable for high-level reasoning. Also, in Chapters 5 and 6 I will present an application that, with no domain knowledge, reconciles the incompatible schemas chosen by designs of databases and semi-structured information sources. Both of these efforts can be seen as work toward opening data sources to semantic interoperation.

### 3.4 Ontology alignment

In the mid-1990s formal ontology became popular in artificial intelligence research, perhaps as a consequence of the rise of interest in intelligent software agents, which tend to encourage the explicit expression of small, self-contained sets of knowledge in a form suitable for reasoning. Although the word “ontology” has traditionally meant the philosophical study of existence (as opposed to epistemology), computer science researchers repurposed the word to refer to a class of structures that are “an explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them”, according to FOLDOC⁵. Probably the most quoted definitions of an ontology in computer science terms are that of Gruber — “an ontology is an explicit specification of a conceptualization” [47] — and Uschold and Grüninger — “an ontology is a shared understanding of some domain of interest” [131].

These two definitions are interesting to contrast, because each includes something that the other omits, and both omissions reveal something of the assumptions that are made when the computer science field appropriates the term. Gruber’s definition requires that there be an explicit specification of whatever conceptualisation an agent (software or human) holds, regardless of whether any other agent also holds an equal

⁵The Free On-line Dictionary Of Computing, accessible at foldoc.org
conceptualisation, while Uschold and Grüninger’s definition requires that the conceptualisation be shared, implying that an understanding of a domain of knowledge is not an ontology unless more than one agent holds it, whether or not the conceptualisation is ever made explicit. Viewed in this way, neither definition is necessarily what might be most appropriate for a particular application, as it is not difficult to imagine on the one hand a scenario in which the constituents of a system share a conceptualisation that is not explicitly specified, and on the other hand a scenario in which each member of a system has its own explicitly specified conceptualisation — indeed, such systems will be the basis of the experiments presented in this thesis.

Regardless of quibbles of definitions, the focus of this thesis is semantic interoperability, and much work has been done to enable interoperability between systems and agents whose knowledge has been explicitly expressed in the form of an ontology. I originally began the work that became this thesis speaking of ontological reconciliation, by which I meant that although agents and groups of agents represent their conceptualisations of the world or a particular domain differently, the real concepts, entities and relationships that their conceptualisations represent are in fact the same, and so any divergence between different parties’ representations, being the ontologies that they design and record, are movements away from a common origin. Thus, the term reconciliation indicates a return to the original concepts. As formal ontologies became popular, the focus of much of the field seemed to shift to the task of ontology alignment, a somewhat lighter task that isn’t concerned with the origins of a group of ontologies, merely accepting that they are different and then seeking any common ground by which they can be “aligned”, permitting interoperability. This turns the problem of semantic interoperability into one that is much closer to the traditional problems of computer science, and can thus be tackled with some of the traditional approaches of computer science, particularly graph manipulation techniques.

The reasons that I am being somewhat critical of this shifting of attention away from what I see as the origins of the problem of semantic interoperability are twofold. Firstly, it seems in some way less urgent to try to reconcile two ontologies if there is no apparent underlying correspondence between the concepts and relationships they represent. If a researcher, sitting in a lab, prepares two similar but different ontologies and then designs an algorithm to align them, what has been achieved other than an interesting exercise in data structure manipulation? Secondly, when the assumption that two ontologies share a common origin is neglected, some information is also set aside that could surely assist in the task of bringing the ontologies back into agreement. Regardless of this criticism, though, the many techniques developed by researchers working with
computer ontologies are undoubtedly useful, and it is appropriate to cover the more notable examples here.

Because the field of computer ontology is quite new, and it draws on concepts from a wide range of related disciplines, it has had difficulty settling on a standard terminology for its activities and goals (vis the multitude of names for very similar activities: ontological reconciliation [66, 67, 68, 69, 70], ontology alignment [33, 22, 62, 107, 58, 35, 36], semantic interoperation [9, 89, 134], database integration [100, 78, 37, 71, 108, 32], data fusion [137, 138, 72, 61], et cetera). Considering the problem of reconciling a pair of explicit ontologies, Hameed, Preece and Sleeman distinguished three activities from the literature [49]:

- Merging is the act of building a new ontology by unifying several ontologies into a single one [102, 126]. The ultimate goal is to create a single coherent ontology that includes all information from all the sources [97]. The new ontology is created from two or more existing ontologies with overlapping parts, and can be either virtual or physical [60].

- Aligning is used when sources must be made consistent and coherent with one another but kept separately [97]. It involves bringing two or more ontologies into mutual agreement, making them consistent and coherent [21, 60]. A set of alignment statements are created during this process, which collectively define the relationships between the original ontologies.

- Integrating entails building a new ontology by composing parts of other available ontologies [102]. Like merging, this process results in a new ontology. The difference between this approach and merging is that only parts of the original ontologies will be integrated — the goal is not to achieve a complete merger.

This degree of fine distinction is useful in that it reassures researchers that work on merging, aligning and integrating is all related and comparable, but beyond this the differences in these definitions seem primarily concerned with application rather than theory and technology. Whether the desired output of an act of ontological reconciliation is a new super-ontology or a set of translations between ontologies seems unlikely to have a significant impact on the techniques that will be effective. In keeping with the literature, when discussing techniques and approaches to reconciling explicit
ontologies I will use the term *ontology alignment* unless the authors of a particular approach themselves choose to use a different phrase.

### 3.5 Assisted and automatic ontology alignment

The theme of this thesis is enabling software systems to automatically reconcile semantic differences in information presentation. One avenue for this advancement is the automation of ontology alignment. In contrast to the systems that merely present ontologies, schemas or service descriptions to users and rely on the user to perform all the ontological reconciliation, there have been a number of attempts in the last decade to automate this process. Automated approaches to aligning ontologies have generally taken one or more of three approaches, dealing with either:

- the *terms* of the ontologies;
- the *structure* of the ontologies; or
- the *instances* of the concepts of the ontologies.

A similarly-styled classification of approaches into *syntactic, external and semantic* techniques is described in [113].

Approaches based on the terms of the ontologies employ linguistic techniques to identify similarities between the names of concepts in different ontologies. These techniques range from simple string comparison functions (see Section 5.3 for a more detailed presentation of string comparison methods), through syntactic operations such as pluralising and stemming, in which concept names are treated as words with a particular grammatical form which are then permuted to find other forms of the same word before comparison so that, for instance, ‘cars’ will match ‘car’ and ‘running’ will match ‘run’, to semantic comparisons which typically involve using a thesaurus to identify synonyms, such as ‘car’ and ‘automobile’. The WordNet project [87, 88] is commonly used as a thesaurus for matching ontology concepts, as it provides both a remotely accessible application programming interface and a function to calculate the ‘semantic distance’ between two words. More sophisticated techniques consider semantic relationships beyond synonymy, most commonly hyponymy, where one concept is a specialisation of another, such as “bicycle” and “wheeled vehicle”, and meronymy, where one concept is a part of another — a more detailed discussion of these relationships and their occurrence in language, as well as their representation in WordNet, can be found in [87].
Term-based approaches to ontology matching are suited to situations where the schema or ontologies to be reconciled are clearly labelled with meaningful terms. Additionally, the terms must be simple — single words or at most a noun phrase. This makes such approaches appropriate for hand-crafted ontologies, such as those presented in academic papers, or catalogue-style taxonomies, such as category/topic filing structures. In practice, such neat concept labels are not always present; abbreviations (e.g. \texttt{adr} for \textit{address}, \texttt{loc} for \textit{location}, \texttt{tel} for \textit{telephone}), prefixes, suffixes and coded tags (e.g. \texttt{s\_name} are commonly used by database designers, as is the practice of joining words, such as \texttt{IsDeleted}, \texttt{RuntimeHours} or \texttt{MailingAddress}\textsuperscript{6}.

Because computer science ontologies are formal structures, they can generally be treated as graphs (such as by placing concepts on nodes and labelling arcs with the relationships between the concepts). Doing so opens the door to applying graph manipulation techniques, the results of which can often be interpreted semantically. For instance, if a concept labelled \texttt{child} is found in two ontologies, and in each ontology the concept \texttt{child} is connected to a second concept by a relationship labelled \texttt{born-of}, but the second concepts in each ontology are labelled differently, it can be argued that the second concepts are probably semantically equivalent.

Finally, the instances of concepts can be compared to establish relationships between the concepts themselves. A concept can be considered to be defined by the set of instances that represent it, particularly when the context of the concept is a tightly-restricted one, such as a database. The third class of ontology alignment approaches involves processing the instances of each concept, in many cases the records of a database but sometimes the explicit instance objects supplied with a formal ontology, and identifying equivalences between the sets of instances. Even though two fields of a pair of schemas might be named differently, if it is possible to find the same values in the instances of both data sources then an assumption can be made that the fields have the same meaning. For example, if there is a field named \texttt{suburb} in one source’s schema and some instance values of that field are \texttt{Brighton}, \texttt{Sandringham} and \texttt{Beaumaris}, then if those same instance values can be found in a field named \texttt{location} from another data source, then it might be safe to assume that the \texttt{suburb} and \texttt{location} fields represent the same meaning. This assumption can be strengthened if the instance values found are not found in any other fields.

\textsuperscript{6}These examples are inspired by a blog entry published at http://weblogs.asp.net/jamauss/articles/DatabaseNamingConventions.aspx — such a publication cannot be considered an authoritative source, but given the number of comments posted on the entry, and the several links to it, I would contend that it is reasonably representative of common practice.
Most approaches to automatic alignment use a mix of terms and structure to align ontologies or schemas, with only very few considering instances. For instances, in 1999, Noy and Musen presented SMART [95], an algorithm for ontology merging and alignment that goes beyond simple class-name matches and looks for linguistically similar class names, studies the structure of relations in the vicinity of recently merged concepts, and matches slot names and slot value types.

They then incorporated this algorithm into a tool called PROMPT [96], which automates the process [of ontology alignment and merging] as much as possible. Where an automatic decision is not possible, the algorithm guides the user to the places in the ontology where his intervention is necessary, suggests possible actions, and determines the conflicts in the ontology and proposes solutions for these conflicts.

Implemented as a plug-in module for the popular Protégé ontology editing tool, PROMPT functions by iteratively making suggestions to the user about how two ontologies might be aligned, acting on the user’s choices and then presenting new suggestions based on the results of the previously requested actions. In this way, it is a semi-automatic approach to ontology alignment that requires regular interaction by an operator. PROMPT works principally by combining two ontologies and then resolving conflicts and redundancies in the new ontology. The techniques used are both linguistic and structural: pairs of classes that have linguistically similar names are suggested for merging into a single class, and sub- and super-classes of these classes are then also compared linguistically based on their relationship to the merged classes. A number of other heuristics are applied to clean up redundant paths between classes and dangling references [96].

Noy and Musen reported that experts who used PROMPT to merge two ontologies followed 90% of PROMPT’s suggestions, including 75% of the conflict-resolution strategies that PROMPT proposed [96]. Additionally, PROMPT was able to suggest 74% of the total knowledge-base operations that the experts invoked during the merging [96].

Chimaera, presented by McGuinness, Fikes, Rice and Wilder in 2000 [75, 77] supports users in merging multiple ontologies and evaluating ontologies with respect to their coverage and correctness. Chimaera analyses the ontologies to be merged, and if linguistic matches are found, the merge is done automatically; otherwise the user is prompted for further action.
TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS

The Semantic Knowledge Articulation Tool (SKAT) prototype system described by Mitra, Wiederhold & Jannink in 1999 [90] used rules expressed in first-order logic to semi-automatically determine matches between two ontologies which can be represented in XML. Rules were formulated to express match and mismatch relationships and methods were defined to derive new matches. The user was required to initially provide application-specific match and mismatch relationships and then approve or reject generated matches. The system included name matching and simple structural matches based on specialisation hierarchies.

In 2004, Rahm, Do and Maßmann surveyed the automatic alignment methods that had been published to date and concluded that they had neglected situations involving very large schemas [103], such as the XML Common Business Library [139], which includes ontologies that contain hundreds of elements. The authors acknowledged the success of other approaches with small schemas and ontologies, and proposed a complementary method that accommodates very large schemas by focussing on types of elements as an extra distinguishing characteristic for matching, identifying schema components (small groups of elements and relationships, such as “street”, “number” and “town”) that have been re-used through the schema, and references within the schemas to external schemas such as type libraries [103]. At the time, the authors did not report any results or evaluation. They subsequently published a revised implementation of their method as a system called COMA++ [7], again without results or evaluation, adding a graphic user interface to facilitate human supervision of the schema matching process. However, in 2006 the COMA++ system was extended to include instance-based matching and entered in the Ontology Alignment Evaluation Initiative contest described in the next section, with notable results [34].

Large-scale schema mapping evaluation has been tackled by Avesani, Giunchiglia and Yatskevich [8]. Beginning with the assumption that schemas can be represented as graphs as described in [43], the authors cleaned and pruned the web site directories published by Google, Looksmart and Yahoo! and then created a set of correct mappings between their entries. They then applied the COMA ontology alignment system described above and the S-Match tool presented earlier by Giunchiglia, Shvaiko and Yatskevich [44]. Rather than a presentation of an approach to mapping large schemas, however, the work presented by the authors is a consideration of techniques for evaluating such approaches.

An, Borgida and Mylopoulos have presented an method for automatically discovering the complex semantics of relational database schemas. Unlike the other methods for enabling semantic interoperation described here, which generally do their best to
propose a mapping between pairs of schemas or ontologies and then rely on the user to exercise the final judgement to approve or reject each mapping, this approach depends on a user supplying what the authors call simple correspondences between fields in a database’s relational tables and concepts in an ontology, which the system described then uses to infer more complex semantic correspondences via a description logic [2, 3, 4]. However, the authors have only reported an informal evaluation of their method, stating in [4] that their implementation system was able to produce correct correspondences for 31 out of 36 sample tables.

3.6 Evaluating ontology alignment methods

More recently, an effort has emerged to provide a standard set of tests to measure the relative effectiveness of different ontology alignment techniques and algorithms. Shvaiko and Euzenat [113], in describing an API for evaluating the performance of ontology alignment algorithms, defined ontology alignment as a set of mapping elements \( id, e, e', n, R \), where:

- \( id \) is a unique identifier of the given mapping element;
- \( e \) and \( e' \) are entities (e.g., tables, XML elements, properties, classes) of the first and the second schema/ontology respectively;
- \( n \) is a confidence measure in some mathematical structure (typically in the [0, 1] range) holding for the correspondence between the entities \( e \) and \( e' \); and
- \( R \) is a relation (e.g., equivalence; more general; disjointness; overlapping) holding between the entities \( e \) and \( e' \).

They then provide a framework that supports the application of different algorithms to a standard set of ontologies. This framework grew out of a series of contests that have been held recently to encourage researchers to compare and contrast their approaches. These annual contests began in 2004 with the EON Ontology Alignment Contest\(^7\), which then evolved into the Ontology Alignment Evaluation Initiative (OAEI) [35]. At the most recent alignment contest, seven sets of ontologies were included in the series of tests to which the entrant algorithms were applied. These test sets were divided into four tracks [35]:

\(^7\)http://oaei.ontologymatching.org/2004/Contest
• **The benchmark track:** Like in previous campaigns, systematic benchmark series have been produced. The goal of this benchmark series is to identify the areas in which each matching algorithm is strong or weak. The test is based on one particular ontology dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain for which alignments are provided.

• **The expressive ontologies track:**
  - **Anatomy:** The anatomy real world case deals with matching the Adult Mouse Anatomy (2,744 classes) and the NCI Thesaurus (3,304 classes) describing the human anatomy.

• **The directories and thesauri track:**
  - **Directory:** The directory real world case consists of matching web site directories (like open directory or Yahoo). It has more than four thousands of elementary tests.
  - **Food:** Two SKOS\(^8\) thesauri about food have to be matched using relations from the SKOS Mapping vocabulary. Samples of the results are evaluated by domain experts.
  - **Environment:** Three SKOS thesauri about the environment have to be matched (A-B, B-C, C-A) using relations from the SKOS Mapping vocabulary. Samples of the results are evaluated by domain experts.
  - **Library:** Two SKOS thesauri about books have to be matched using relations from the SKOS Mapping vocabulary. Samples of the results are evaluated by domain experts.

• **The conference track and consensus workshop:** Participants have been asked to freely explore a collection of conference organization ontologies (the domain being well understandable for every researcher). This effort was expected to materialize in usual alignments as well as in interesting individual correspondences (“nuggets”), aggregated statistical observations and/or implicit design patterns. There is no *a priori*

---

\(^8\)The SKOS (Simple Knowledge Organisation System) effort is a project under the auspices of the World Wide Web Consortium (W3C). It aims to develop “specifications and standards to support the use of knowledge organisation systems (KOS) such as thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web” — see http://www.w3.org/2004/02/skos/intro/ for more details.
reference alignment. Organizers of this track offer \textit{a posteriori} evaluation of results in part manually and in part by data-mining techniques. For a selected sample of correspondences, consensus will be sought at the workshop and the process of its reaching will be recorded.

Although a few of the test ontologies include instances, the major focus of the contest is on textual and structural aligning of ontologies, with relatively little attention paid to instance-based alignment, and this is reflected in the algorithms that are entered into the contest. Although the contest has attracted more entrants and has presented harder challenges each year, many entrants do not attempt the more difficult tests. Figure 3.4 shows the entrants to the contest in 2007 and which of the tests they attempted [35].

<table>
<thead>
<tr>
<th>Software</th>
<th>confidence</th>
<th>benchmark</th>
<th>anatomy</th>
<th>directory</th>
<th>food</th>
<th>environment</th>
<th>library</th>
<th>conference</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgreementMaker</td>
<td>√√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOAS</td>
<td>√√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASMOV</td>
<td>×√××</td>
<td>×√×××</td>
<td>×√×××</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSSim</td>
<td>×√×××</td>
<td>×√×××</td>
<td>×√×××</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Falcon-AO v0.7</td>
<td>×</td>
<td>×√×××</td>
<td>×√×××</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lily</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLA2</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OntoDNA</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OWL-CM</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior+</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RiMOM</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sambo</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCARLET</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEMA</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Silas</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TaxoMap</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X-SOM</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong>=17</td>
<td>10</td>
<td>13</td>
<td>11</td>
<td>9</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

\textbf{Table 2.} Participants and the state of their submissions. Confidence stands for the type of result returned by a system: it is ticked when the confidence has been measured as non boolean value.

\textbf{Figure 3.4:} Entrants in the OAEI ontology alignment contest in 2007 [35]
A more unusual approach to ontology alignment was described by Schaaf in 2002 [106], in which an approach to reconciling multiple organisational memories is presented as an exercise in aligning ontologies. As used by Schaaf, the term “organisational memory” refers to the information stored in systems that support activities such as storage, retrieval and sharing of explicit knowledge in an organisation. In the case presented, explicit knowledge meant information that has been expressed in terms of an explicitly specified conceptualisation, or ontology [106]. However, no results were reported by Schaaf.

Perhaps the research most similar to either of the three experiments presented in this thesis is the Learning Source Descriptions (LSD) system described by Doan, Domingos and Halevy in 2001 [29], which appears to have been extended to create the iMAP system presented by Dhamankar, Lee, Doan, Halevy and Domingos in 2004 [28]. Like the AReXS system presented in Chapter 5, the iMAP system semi-automatically identifies matches between fields in pairs of structured or semi-structured data sources [28], although the iMAP system goes further and identifies what the authors describe as “complex matches”, in which a field in one data source is semantically equivalent to a combination of two or more fields from another data source. An example given by Doan and his co-authors is that a field labelled \texttt{list-price} from one source contains the same information as the result of combining two fields, \texttt{price} and \texttt{fee-rate}, via the formula \texttt{price} \times (1 + \texttt{fee-rate}).

The iMAP system is based on examining instances from data sources, which makes it somewhat unusual in the field of data scheme and ontology reconciliation. To identify potential matches between fields, a Naive Bayes text classifier is trained on the contents of one field as found in the instances of one source and then the contents of each field in turn from the second source are examined by the classifier to decide which field from the second source is best predicted by the field from the first source [28]. This process is then repeated for combinations of fields, including concatenations and more complex combinations as mentioned above.

The authors claim that these complex matches are important, and they present four domains in which they report having found a large number of such matches between sample data sources: databases of houses listed for sale, product inventories for grocery stores, cricket players and financial data of clients [28].

In addition to considering complex matches, the iMAP system is notable for its use of heuristic functions for comparing instances. The system includes functions for comparing instances textually via string comparison, numerically, as categories, via unit conversion, as dates and by schema mismatch recognition. Each of these heuristic
functions appears to be manually implemented and provided to the system as a result of a person observing the data to be reconciled and proposing a non-textual relationship. Adding such capability certainly increases the effectiveness of the iMAP system, but it is not automatically scalable as each new heuristic requires human intervention. In this way this aspect of the iMAP system is similar to the earlier manual attempts at data integration described earlier in this chapter. However, the authors of the iMAP system have provided another interesting feature to reduce this dependence on human intervention: one class of heuristic involves the construction of arithmetic combinations of fields, and a specialised search module is able to generate candidate combinations from a provided set of operators, such as addition, subtraction, multiplication and division. In this way, the iMAP system can discover relationships such as the \( \text{price} \times (1 + \text{fee-rate}) \) function described above by itself, without requiring a person to have previously considered it. This is quite a significant advance, although the authors deliberately constrained the search module to simple functions as the number of possible functions that could be explored is obviously unlimited. Again, one limitation is that the operators themselves must be pre-defined by a person, although it seems unlikely that this is really much of a problem.

One aspect of the iMAP system that is not clearly explained is how the data instances are actually compared textually. The authors state that scores are computed based on similarity, but the actual comparison function is not defined\(^9\) (see Chapter 5 for a discussion of common string comparison methods).

The iMAP system uses further techniques to improve its results, such as pre-analysis of the data found in fields to guess ranges of possible values and eliminating proposed matches if one of the involved fields is also present in another higher ranked potential match.

The evaluation presented by the authors involved obtaining a database of real data from each of several domains to serve as a source schema. For each domain, volunteers were then asked to create a new schema to serve as a target \([28]\). It is not clear what criteria or instructions were given to the volunteers, nor is it clear how the target database was populated with instances. Finally, volunteers were asked to examine the pairs of source and target schemas and identify one simple or complex match for each field in the target schema \([28]\). The iMAP system was then given the pairs of databases, for which it proposed a set of field matches. According to the authors, the iMAP system proposes a field match for every field in the target schema, although it is not clear why this is the case — this implies that no target schema contained any fields that were not

\(^9\)A reference is made to the thesis of one of the authors, but I have been unable to find this document.
present in the source schema or could be generated by combining fields from the source schema, which makes the target schemas seem unrealistic. Regardless, this means that precision and recall measures are not necessary, as the recall is always perfect. Instead, the authors present the matching accuracy of the proposed field matches, calculated as the percentage of proposed matches that were deemed correct by comparison to the volunteers’ choices, which seems to be equivalent to the standard precision measure.

The authors report two sets of results, one for the case where the source and target database contained some common instances and another for the case where no instance was represented in both databases. For the latter case, the matching accuracy was found to be 27-48%, and for the former case, the accuracy was 50-86%. As the authors note, when the source and target databases share no common instances, matching by finding the representation of one object or concept in both schemas or data sources is effectively impossible, and so the only success possible is by co-incidental similarities between different objects. In certain circumstances, however, this is not a critical limitation, as will be discussed in Chapter 6.

Having presented a wide range of approaches and considered the strengths and weaknesses of each, it seems highly likely that no one technique can solve the problem of semantic interoperability. This thesis will present three explorations of the problem, drawing on the approaches and techniques presented in this chapter, and extending them as necessary. Although a hybrid approach combining heuristics, logic, statistics and domain knowledge seems essential, this doesn’t mean that it is not valuable to work to improve a single technique in isolation, as any observed improvements can be clearly ascribed to the particular technique, and all improvements made can surely be effectively applied to future combined approaches. The three studies that form the contribution of this thesis are an integration of a massive knowledge-base expressed in formal logic with a lightweight heuristic-driven information agent and a semi-structured data source, an exploration of the limits of an instance-based algorithm for reconciling the schema of structured data sources, and a pre-implementation attempt to reduce the problem of semantic interoperability by allowing developers to establish a decentralised consensus about the representation of data and information. The explorations will be described in detail with the hope of providing a foundation for future researchers and developers to build upon, and where possible, standard evaluation measures will be applied to aid in comparison of the results with other work. Where no appropriate evaluation measure can be determined, a critical reflection will be provided and I will endeavour to openly identify what I believe are the strengths and weaknesses of the work.
Chapter 4

Tasks, Context, Cyc and Sports Reports

This chapter describes several stages of progress toward a general framework for building and eventually automatically generating software agents to supplement the Cyc knowledge-base with information from external sources\(^1\). An information agent called SportsReporter will be the basis for this exploration, and it will undergo several significant re-factorings during the story.

As discussed in the previous chapter, when general ontologies and knowledge-bases grow in number and size, the issue of when certain statements or assertions are true and relevant and when they are not becomes more important. Legg gives examples of simple statements that require extra information in order to determine their truth or meaning by identifying logically contradictory statements that might be encountered while reading books, newspapers or web pages [64]: the assertion “New Orleans escaped major hurricane damage” could be stated truly prior to August 31st, 2005 but not after, demonstrating the need to consider the temporal context of an assertion, and the pair of statements “John Brown was born in 1945” and “John Brown was born in 1967”, which cannot both be true unless the entities indicated by the textual sign “John Brown” are different, in which case context, in the form of a definitive identification of which particular entity is relevant, is essential to resolving the apparent inconsistency of the statements.

It is important here to state clearly what I mean when I say that a statement is ‘true’ or ‘false’. If you will pardon the analogy, truth is like pornography, in that it

\(^1\)The work presented in this chapter was supported by an ARC Linkage Grant provided by the Australian Government and Cycorp, Inc.
can be difficult to define but we are confident that we know it when we see it, and the history of philosophy is replete with competing theories, definitions and explanations of what is meant by the concept. Since computer science deals almost universally with abstract models, rather than the real world, and even computer ontologies typically do not attempt to define the relationship between their constituent concepts and the external world, it seems to me that it is reasonable to adopt an understanding of ‘truth’ that is rooted in common or lay usage. The correspondence theory of truth has been considered to be analogous to the common interpretation of the idea, namely that a statement is true as it corresponds to a fact, or to reality [27]. I will maintain this interpretation for the remainder of this chapter.

I will also take as a basis for this discussion the notion that no statement can be considered to be true or false without reference to a context that completes the statement and resolves any ambiguity that might be present. One could consider ‘reality’ to be the default fallback context, which is to be assumed when no other context can be determined. I will argue that statements are typically incomplete and are generally made with a particular context assumed, and that sometimes this context is explicit and sometimes it is implied. This can lead to confusion, where the reader or hearer of a statement is uncertain as to which context was intended, and is therefore unable to accept the statement, or misunderstanding, in which case the recipient assumes a context different to that intended, and thereby (incorrectly) accepts the statement as having a different meaning.

I began this chapter by quoting two examples of statements which require a context to correctly determine their meaning and truth. I should emphasise that I am not suggesting that reality or facts are subjective; rather, that the statements that we make and receive are generally intended to be interpreted within a particular context, that that context is often implied, and that the meaning of the statement changes depending on the context in which it is interpreted. Further, a statement may, when completed by at application of one context, be true while the same statement, interpreted in another context, may be false. Given that the theme of this thesis is communication and interaction between people, software and combinations of the two, I also intend to allow for a fairly loose understanding of ‘reality’ — because people’s knowledge of reality is often flawed and arguably always incomplete, it is helpful to be able to refer to the ‘truth’ of statements that they make or perceive in relation to their knowledge of reality, rather than reality itself.

In addition to the temporal and referential contexts illustrated in the first two examples given above, other forms of context are also important in completing a statement
and thus determining its truth. For example, the statement that vampires exist would generally be considered untrue in the context of contemporary Western culture because in reality such creatures do not exist\(^2\), yet this culture also contains a widely-known genre of fictional stories in which creatures known as vampires are very real. Although considered to be obviously fictional by almost everybody who lives in contemporary Western culture, such a statement will certainly be encountered by a computer program reading the World Wide Web, and thus such a machine will require the ability to contextualise in order to avoid confusion.

In general, in different contexts, the ‘truth’ of a statement will be different. Further, in addition to deciding whether a statement is true or not, most users of information will also want to evaluate it based on subjective criteria such as importance, relevance, urgency or reliability. Software systems will increasingly be expected to deal with ordinary, real information as we ask them to assist us with more and more everyday tasks. If knowledge models and ontology-based applications are to be widely usable by large numbers of people and software agents then it is necessary to accommodate this subjectivity.

Given the difficulty of modelling and representing knowledge and reasoning in the first place, it is not clear how to extend such models and representations to deal with subjectivity. The idea usually employed as an aid in approaches to extending models is that of context, that the meaning of a word or text is determined by the texts and statements that surround it. Additionally, context incorporates the circumstances or setting in which an event occurs. For example, if a news report includes a local temperature forecast of 25 degrees, it is reasonable to act on that knowledge and choose clothing or make plans accordingly; but if the report was generated in a foreign country, it would no longer be sensible to base your choices on the information that it contains. Although the statement “the temperature will be 25 degrees tomorrow” is bound to be true for people in certain places, it is most unlikely, barring coincidence, to be true for us — the statement is incomplete, and if we complete it with a context other than one on which is was intended to be interpreted, the result will be false.

The question of whether the forecast temperature has been given in degrees Celsius or Fahrenheit is also an example of the importance of context for correctly interpreting information. Without some extra information, such as the country of origin of the forecast and which scale of temperature is generally used there, it is very difficult to decide whether tomorrow will be quite warm or very cold. Although this might seem

\(^2\)As far as we know — here lies the rub in being too strict about what is real and true, in that almost every statement really requires the qualification “as far as we know”.

51
trite, it is exactly this sort of contextualising information that is often omitted when information is recorded in software systems, as I discussed in Chapter 1. One simple example of this is the Mars Climate Orbiter spacecraft, which was lost in 1999 due to one component of the control system passing data in imperial units and another component treating the data as though it was metric [92]. Another example is the tendency of stores to publish prices for items on their web site, but fail to indicate the currency of the value — if a reader makes the assumption that the prices are expressed in their own currency, thus completing the completing the statement by interpreting it in a context, then the resulting statements generated by different users will sometimes be true and sometimes false.

The example of a temperature forecast needing a context to be meaningful bears examination with a view to what exactly it is that renders a single piece of information relevant or irrelevant, useful or useless. In the case of the weather report, it is the presence in the surrounding information, or context, of a fact that indicates that the report originated in a place sufficiently removed from your current location that the contents of the report are not applicable to you. A key observation from this example is that rather than asking about the meaning of an assertion, we could ask about its relevance to us, in our current situation, given our current activities and goals. The relationship between meaning and relevance is not completely clear; it can be difficult to say whether the meaning or the relevance of an assertion is more valuable. Arguably, the meaning is useless without some consideration of the relevance, while relevance would seem to be difficult to determine without an idea of the meaning. In the above example of a weather report, the interests of the recipient of the report viewer are served equally well by judging the forecast irrelevant based on its context as they are by considering its meaning in light of the origin of the report — if the former can be done quickly, it seems unnecessary to perform the latter.

In many situations, context extends beyond the contents of a particular information source or report. Motivation, immediate and future goals, attitude and available resources all tend to influence the interpretation of both the truth and the relevance of information. In general, the meaning and the relevance of a piece of information are bound to be dependent on not just the obvious context — the extra information surrounding a particular piece of data — but also whatever facts are already known or believed: even if a weather forecast is judged to be applicable to your neighbourhood, if you have no intention of going outside at all then the report has little relevance. In group environments, individuals will interpret information differently according to the roles they are playing and the tasks they are performing. For example, to the comman-
der of an entire theatre of war, the particular formation of a group of enemy aircraft is not at all relevant — what matters is their apparent destination. However, to the pilots tasked to engage the enemy aircraft, the formation may be a crucial factor in choosing which tactics to employ. Similarly, to the average person it is true that it is impossible for something to jump instantly from one place to another; simple Newtonian physics and their everyday experiences rule out anything else. But to a physicist, quantum theory reverses that truth. Likewise, software agents in multi-agent systems increasingly have to deal with different interpretations and viewpoints, as theories based on ideas such as joint intentions, dynamic groups and hierarchies, collaborative planning and heterogeneous communities are becoming increasingly popular, as can be seen by the number of conferences, workshops and tracks that cover these topics: AAMAS\textsuperscript{3}, AAWSE\textsuperscript{4}, ASAMI\textsuperscript{5}, EUMAS\textsuperscript{6}, IASB\textsuperscript{7}, IDC\textsuperscript{8}, IWEA\textsuperscript{9} and KR\textsuperscript{10}, to name a few.

The Cyc project is one of the longest running projects in the field of artificial intelligence, and its handling of context in determining meaning makes a good starting point for this discussion. In this chapter, exploring the role of context and meaning in enabling the performance of tasks will lead to the introduction of a software agent built to answer the age-old question: did my sporting team win on the weekend? The core of the chapter is a presentation of efforts to integrate the lightweight, task-oriented software agent with the massive, general-purpose Cyc knowledge-base, and the issues that arose while trying to fit a domain-specific ontology into a larger global ontology while retaining meaning. Further, the roles of the lightweight software agent and the massive knowledge-base are then reversed as the heuristic knowledge embodied in the agent is transferred to Cyc, with the eventual goal of providing Cyc with the ability to generate new forms of the information agent on demand to answer different questions in other domains.

\textsuperscript{3}International Conference on Autonomous Agents and Multiagent Systems — www.aamas08.org  
\textsuperscript{4}International Workshop on Agents for Autonomic Web-Service Environments — users.encs.concordia.ca/ bentahar/Workshop  
\textsuperscript{5}Symposium on Artificial Societies for Ambient Intelligence — asami08.cs.rhul.ac.uk  
\textsuperscript{6}European Workshop On Multi-Agent Systems — www.atia.rnu.tn/eumas  
\textsuperscript{7}Society for the Study of Artificial Intelligence and the Simulation of Behaviour — www.aisb.org.uk/convention/aisb08  
\textsuperscript{8}International Symposium on Intelligent Distributed Computing — idc08.diit.unict.it  
\textsuperscript{9}IEEE International Workshop on e-Activity — www.iwea.net  
\textsuperscript{10}International Conference on Principles of Knowledge Representation and Reasoning — kr.org/KR2008
4.1 Introducing Cyc

The long-standing, highly-visible Cyc project is a useful study in the practical implementation of context in a functioning ontology-based system [23, 63]. Cyc is interesting because its knowledge-base/ontology is huge and general — it attempts to be completely objective in the hope of being understandable and reusable by anyone. The Cyc project is a very large, multi-contextual knowledge-base and inference engine, currently containing more than 300,000 concepts and 3 million assertions [23]. Concepts can be marked as specialisations of other concepts, creating a tree-like structure of inheritance relationships that allow existing assertions to be applied to new knowledge without being specifically defined. Cyc knowledge is represented via a predicate logic language called CycL, and Cyc itself is implemented in a variant of Common Lisp called SubL. Cyc can also contain meta-facts such as functions that extend itself by processing other facts and rules, as well as functions that allow it to interact with external systems.

A project has existed within Cycorp for several years to connect the Cyc knowledge-base to external data sources, although it has so far been primarily concerned with structured databases rather than semi- and unstructured information sources such as agents and web sites. This effort, known as the Semantic Knowledge Source Integration project (SKSI) [81, 82], has provided Cyc with the ability to query external databases, web sites, and other applications, translating the information contained within them into atoms and predicates that can then be reasoned about alongside Cyc’s own hand-crafted and inferred knowledge. The SKSI project aims to go beyond manual translations and wrappings of databases and actually describe the structure of an external information source in Cyc, so that Cyc itself can consider the contents of the source and decide how best to integrate them with it’s own knowledge.

The primary motivation for the project is to make the myriad tiny facts stored in databases and other record stores available to Cyc, without requiring a person to laboriously enter their minutiae. It would be valuable for Cyc to know the daily recorded temperatures for major cities or the daily index values of major stockmarkets, but to add all these facts into Cyc would require either manually asserting every value, which is almost certainly never going to be undertaken, or creating a customised importing script or program to dump the records as at a particular time and generate Cyc assertions for each value, which is neither elegant nor efficient. Such an approach to interaction between two information systems would be at a much lower level than most

\[\text{It seems that the SKSI project was originally titled} \ Structured \ Knowledge \ Source \ Integration \ but \ more \ recent \ references \ to \ the \ project \ call \ it} \ Semantic \ Knowledge \ Source \ Integration.\]
software engineers would consider appropriate, but worse is the fact that the resulting information would immediately be out of date, as new records are created every day. Thus, the approach taken by the SKSI project is to provide Cyc with the mechanisms to query databases and other information sources, and the knowledge to reason about how to interpret the resulting data.

The information agent that I will present in this chapter has been implemented via the SKSI API, in a fashion similar to several other extension modules implemented by Cyc engineers to demonstrate the SKSI project\(^\text{12}\). One notable difference is that the information agent was constructed from an agent-oriented, heuristic-based viewpoint with a view to re-usability and multi-applicability, whereas the internally developed extension modules are wrappers and scrapers designed directly for a particular source. I will mention more details of the SKSI project and framework later in this chapter.

### 4.2 Context in Cyc

During the Cyc project several approaches have been taken to context. The initial direction of the Cyc project was to amass the vast collection of facts that form general knowledge and common sense by explicit specification, in essence compiling the contents of an encyclopaedia. When that endeavour appeared unlikely to be sufficient for intelligence to emerge, the Cyc direction expanded to specifying the larger body of knowledge that surrounds and informs our understanding of the facts in the encyclopaedia, without which we are left struggling to communicate in a sea of knowledge; as Brian Smith put it, “everything we know, but have never needed to write down” [117]. In other words, the daily experiences and mundane common knowledge that permits us to function in the world and to understand the information we read in the encyclopaedia. The purpose of this section is not to critique the Cyc project or its goals, but instead to observe its experiences with the implementation of context-sensitive knowledge manipulation to provide a background discussion for the work presented later in this chapter.

For the first half dozen years of its life Cyc contained no explicit consideration of context [117, 65]. From 1989 to 1991, contextualisation was added to the various attributes or characteristics that concepts and assertions possess in Cyc. Contexts were defined and said to have assumptions and contents, assertions could be imported from one context by another, and contexts were actual first-class terms in the Cyc representation language that were partially ordered by specialisation. Unsatisfied with this,

\(^\text{12}\)Based on personal communication with Cyc developers.
Doug Lenat, the initiator and leader of the Cyc project, listed the primary drawbacks as:

- the expense of importing assertions from one context to another,
- the burden on the ontology builders of explicating the assumptions of each context,
- and the cost of placing every assertion into the proper context [65].

Two things can be learnt from this episode in the life of Cyc: firstly, that some direct consideration of context is necessary for a large ontology-based system, and secondly, that representing context as just another piece of knowledge is not an effective way to capture the particular effects that context has on information. Cyc now has a more sophisticated mechanism for explicating and manipulating context using microtheories\(^\text{13}\) (in the Cyc glossary, the entry for the term context says “often used interchangeably with microtheory” [24]).

Microtheories are based on the idea that context is a multi-dimensional space [65]. The dimensions of this space reflect the attributes of an assertion that define under what conditions it is valid. The twelve dimensions of context-space proposed for Cyc by Lenat in [65] are:

- absolute time,
- type of time,
- absolute place,
- type of place,
- culture,
- sophistication/security,
- topic/usage,
- granularity,
- modality/disposition/epistemology,
- argument preferences,

\(^\text{13}\)Based on personal communication with Cyc developers.
• justification, and

• lets.

By specifying segments along the axes of some or all of these dimensions, regions of the context-space are carved and a context is defined. Assertions in the Cyc knowledge-base can have the appropriate region of context-space specified for them, and queries can be restricted to certain regions of the context-space, thus limiting the set of assertions that can be referenced when Cyc tries to supply an answer. Specified regions of context-space can be reified within the Cyc knowledge-base and then re-used for a set of assertions — these are the actual microtheories.

In practice, microtheories are used as grouping devices that tie together a number of assertions under a label [63]. Those assertions must be true in the context referenced by the label of the microtheory. Thus, each assertion, or truth, in the Cyc ontology is tagged with the microtheory, or context, in which it is true. This opens the way not only for the inclusion of assertions that are apparently contradictory when taken out of context (e.g. using microtheories it should be possible to add to Cyc the two assertions “Jesus Christ was the son of God” and “Jesus Christ was an ordinary man who was a wise teacher”, which cannot both be true in the same context), but also for a calculus of microtheories, allowing various manipulations of not just assertions but whole contexts at a time. Importantly, queries to Cyc are also tagged with a microtheory, effectively adding to the questions an attempt to specify the context in which each question is being asked. As a hypothetical example, when asking the question “Was Jesus Christ the son of God?”, one could specify the microtheory AtheistDoctrineMt and be told by Cyc “No, Jesus Christ was not the son of God”. If, however, the microtheory label provided with the query was ChristianDoctrineMt, the answer would be “Yes, of course”.

Similarly, the roster of teams competing in the F.A. Premier League changes each year due to relegations and promotions, yet it would be useful to be able to assert facts about the teams in the league, and so microtheories could be used to contextualise such assertions. Microtheories could be created to represent each iteration of the league, such as FAPremierLeague2005-06, FAPremierLeague2006-07 and so on (the Premier League doesn’t follow the calendar year). Following this, assertions such as (winner FAPremierLeague ArsenalFC) could be contextualised by specifying a microtheory of FAPremierLeague1997-98, and similarly general queries could be disambiguated in the same way: (winner FAPremierLeague ?) in the context of FAPremierLeague2004-05 could return ChelseaFC.

A more flexible although potentially more complicated way to specify a temporal
context would be to use the appropriate units to precisely select the desired period, such as the following for the F.A. Premier League season 2004-05:

\[
\text{(TimeIntervalInclusiveFn (MonthFn August (YearFn 2004)))} \\
\text{(MonthFn May (YearFn 2005)))} \\
\text{(winnerInConflict FAPremierLeague ArsenalFC)}
\]

In order to be integrated with the Cyc knowledge-base, the new information agent, SportsReporter, needed to be implemented in SubL. The Cyc architecture

The microtheories implementation in Cyc permits hierarchies of microtheories, with inheritance of properties from parent contexts to specialising child contexts, as well as providing for relations to assume default and inherited contexts. A stated aim of the framework is to, by breaking the huge Cyc knowledge-base into contexts, speed both knowledge entry and inference. The need is removed for the knowledge engineer to specify myriad assumptions on which each truth that they enter hinges. Additionally, opportunities are provided for an inferencing engine to efficiently remove large chunks of the knowledge-base from consideration based on what is known about, for example, the context of a query. To paraphrase Lenat, when asking if, given that it is raining, should one take an umbrella, it is not worth considering the number of legs on a spider or the birth date of Julius Caesar [65].

The current Cyc approach to handling and exploiting context looks promising, and produces effective demonstrations, but it is not yet clear whether it can achieve the goals desired. Lenat has explained the choice of twelve dimensions, but is not certain that his choices will be comfortable or even comprehensible for others who wish to use Cyc. He admits that there is no real limit to the number of dimensions of context-space that could be identified and that each dimension is almost certainly continuous rather than discrete, which further complicates the bounding of context spaces by requiring hard delineations. More problematic, though, is the question of how many actual contexts are enough. Although they seem to greatly simplify and reduce the number of assertions required to resolve complex questions of meaning and truth, already Cyc has thousands of microtheories, and there seems to be a clear preference on the part of people adding to the Cyc knowledge-base to name their contexts and work with them as labels, rather than specifying in detail the location of each assertion in context-space\textsuperscript{14}, which significantly reduces the ability of the Cyc engine to reason about contexts, one of the driving motivations for the approach in the first place.

\textsuperscript{14}Based on personal observations of the Cyc knowledge-base.
To illustrate this problem, consider that Cyc knows that geological changes take place over millennia and are generally not observable from decade to decade and century to century. If Cyc is also told that an assertion that a certain mountain exists is true in a microtheory defined as the time period from 1700 AD to 1799 AD, then it can infer that the mountain’s existence will also be true in any microtheories defined for the periods from 1600 to 1699 and 1800 to 1899. On the other hand, if Cyc had been told that the assertion that the mountain exists was true in the microtheory *The 18th Century*, with no further definition provided for that microtheory, Cyc will be unable to infer that the mountain should still exist in *The 17th Century* or *The 19th Century*. Unlike humans, who readily parse and interpret labels and infer properties from their names, Cyc cannot and doesn’t know anything about *The 18th Century* until it is defined as a region of context-space.

It seems likely that the number of microtheories will continue to grow rapidly, particularly as the Cyc knowledge-base specialises. As a small example of the degree of granularity that can be expected to be required, the ACM Computing Classification System [6] classifies technical papers in the area of computing into 1474 topics, in a hierarchy four levels deep. Perhaps Cyc’s general knowledge will make it unnecessary to specify all of these contexts, but even starting from such a specialised area as Computing, it takes a further four levels to specify Object-oriented languages (see Figure 4.1) and that doesn’t seem like an over-specialisation for the context of a query. And when other fields are considered, such as medicine, science, engineering and folklore, the number of contexts appears to grow without limit. Possibly this will not be a problem at all once Cyc understands natural languages and is configured to take advantage of large grid-computing resources. However, given Cyc’s rate of progress so far, this could be a long time coming.

### 4.3 The task at hand

In his exhaustive exploration of the world of 12-dimensional context space, Lenat makes a significant remark that context enables people to, among other things, “ignore 99.999% of our knowledge so that we can focus on the task at hand” [65].

The problems associated with capturing complete knowledge have lead many to limit their efforts to domain-specific approaches to ontology-based systems. Limited-focus knowledge-bases or ontologies have been referred to as purposive, a term that neatly expresses the goal of specifying all and only the knowledge required for a particular purpose or task. So when Lenat writes that context is some intangible thing that
1. General Literature

2. Hardware

3. Software
   (a) GENERAL
   (b) PROGRAMMING TECHNIQUES
   (c) SOFTWARE ENGINEERING
   (d) PROGRAMMING LANGUAGES
      i. General
      ii. Formal Definitions and Theory
      iii. Language Classifications
         • Applicative (functional) languages
         • Concurrent, distributed, and parallel languages
         • ...
         • Object-oriented languages
      iv. Language Constructs and Features
   (e) OPERATING SYSTEMS
   (f) ...

4. Data

5. Mathematics of Computing

6. Information Systems

7. ...

**Figure 4.1:** An extract from the ACM Computing Classification System hierarchy of topics [6]
permits us to identify the set of our knowledge that is not relevant to our current purposeful activity, the conclusion can be drawn that context, practically considered, is defined as much by the purposive knowledge that has been included as relevant to the task at hand as by the knowledge that has been excluded. In other words, when trying to understand what makes some knowledge relevant to a task and what gives it specific meaning for that context, don’t set aside the knowledge identified as necessary to the current purpose and looking for the context in the vast remainder of our knowledge. Instead, look to the defining characteristics of the knowledge already deemed relevant and seek the context within it; if necessary, identifying and making explicit any knowledge assumed or implied by the task definition itself.

In many ontology-based systems that perform information-oriented tasks, purposive knowledge-bases or ontologies provide an efficient way to specify and represent the knowledge needed to successfully operate in the required environment. Lightweight techniques such as knowledge unit analysis [41, 123] attempt to explicitly model common sense knowledge for a strictly limited domain, without trying to use a general purpose formal knowledge representation technique such as description logic. The lightweight approach avoids the mysticism with which context sometimes seems to be viewed, by considering the context to be inherently captured as the purposively relevant domain knowledge units are specified. Put flippantly, if you have defined the task adequately that it can be successfully performed in the desired circumstances, you have surely defined the context sufficiently, or else how is the task being performed?

Even if it is not explicitly discernible, sufficient context can be determined to be present by considering whether the task is being performed correctly. Knowledge unit analysis works, to a degree, at the sub-concept level, by assuming that where there are knowledge units in a text, the concepts that the knowledge units represent are present, rather than trying to decide if the concepts themselves are present or not [123]. In this way, because concepts are not formally defined and generalised with theoretically complete assertion and axiom systems, the problem of formally locating those which are not concepts or assertions but elements of context is avoided. The context is explicated, but it is not discerned from the concepts and other elements that comprise domain knowledge. The close relationship between information and context is due primarily to the importance of the environment in defining and performing a task. The environment in which a task is to be performed plays a crucial part in the definition of the task, often implicitly. The crucial nature of the environment is true for tasks performed both by people and by information agents.
TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS

Score format: \(<\text{number}> - <\text{number}>\)

Team name: (a sequence of) \(<\text{capitalised string}>\)

Winning team: the team with the numerically greater score

Score correspondence: the left-most team is given the left-most score, regardless of the ordering of actual score numbers and team names

Conceivable scores: \(0 \leq \text{potential score} \leq 13\)

Irrelevance: anything between parentheses, e.g., given the information \(\text{Inter Milan} 2 (\text{Baggio} 12, \text{Beresi} 67) \text{ def Lazio} 1 (\text{Simone} 34)\), SportsFinder does not report that Inter Milan lost to Baggio 12 goals to 2.

Figure 4.2: Examples of knowledge units used by the \textit{SportsFinder} information retrieval agent

4.4 SportsReporter

For the rest of this chapter, the role of a task-oriented information agent will be played by SportsReporter, the successor to a lightweight information agent developed in the Intelligent Agent Lab at the University of Melbourne and called SportsFinder. SportsFinder was the result of an experiment to retrieve the results of a sporting match from a web page without having to be told exactly where and how the match result is reported [74, 122].

Developed in an evolutionary manner, SportsFinder showed not only significant success in assessing a given web page to interpret its contents and highlight the information about a certain team, but also remarkable adaptability. Developed to report the outcomes of soccer matches, it was adapted to a range of team sports including basketball, rugby and Australian Rules football. The SportsFinder agent was able to cope well with new information sources, and was able to incorporate new sports on the fly by the user filling out a simple template. Despite — or perhaps due to — the fact that it understood nothing about natural language, SportsFinder coped equally well with the Italian Serie A results as it did with the FA Premier League, and it was also adapted to report on the results of Chinese matches\(^{15}\). The knowledge units that SportsFinder dealt with included those in Figure 4.2.

So where is the context? The point of describing the SportsFinder information agent

\(^{15}\)The adaptation was performed in 1999 by Tiechuan Wang as a directed study project while studying in the Department of Computer Science and Software Engineering at The University of Melbourne.
is to show the success that a lightweight, purposive knowledge-base or ontology gave, despite the clearly high dependency on context of the task of being able to identify which words and number on a sport results page are actually teams and scores, and which teams and scores together represent the outcome of an actual match. This example has clear relevance to the issue of having to rely on context to interpret a text — does the text “3” represent the number of goals scored by a team, the minute in which one goal was scored, the day or month of the fixture, the number of a player dismissed from the field for foul play, the minute in which he was sent off or the position on the league ladder one of the teams now occupies as a result of losing the match? Obviously the immediately-surrounding text defines the meaning of the number. How much of the surrounding text is required? SportsFinder deliberately started with the minimum consideration necessary to solve the problem of reporting match results in one instance, and slowly grew in knowledge until it could handle a very wide range of soccer results information sources.

It is not, however, necessary to build up the required knowledge units slowly on an informed trial and error basis. The conversion of SportsFinder from English to Chinese match reports demonstrated that it is possible to efficiently identify the knowledge units for a new domain that will permit the agent to migrate with success. This is especially encouraging, as it implies that newly created agents could be informed from their beginning with the results of knowledge unit analysis of their intended domain and environment. I want to emphasise that people can and do perform tasks in a corresponding manner to the behaviour exhibited by SportsFinder. My doctoral supervisor tells of enjoying the challenge when travelling internationally of looking up sports scores in local newspapers. He claims to be usually successful despite the wide variety of formats of the newspapers and the fact that he rarely understands the local languages. The task of locating the scores is made achievable by its context. For example, the assumption that the scores will be present explicitly in a certain part of the newspaper makes it possible to ignore large parts of knowledge, such as the syntax of foreign languages.

### 4.5 Meaning and relevance

The knowledge units used by SportsFinder, and the heuristics by which they were generated, raise interesting issues concerning meaning and relevance. At first glance, it is satisfying to imagine that SportsFinder is at some level determining the meaning of individual elements of information, or knowledge units. Certainly, it seems appropriate
to consider the information agent analysing a line that contains several numbers and deciding that this number is probably not a score and that that number most likely is. From this perspective, the meaning of the numbers has been determined and they have been ruled out as candidates for inclusion in the final answer to the user’s question about how their team fared. But an alternative consideration could view the actions of SportsFinder as assessing the relevance of each knowledge unit to the original task. As elements of the web site being processed are deemed unlikely to warrant inclusion in the final response, they are removed. This is where domain-specific knowledge such as “content within parentheses is not a score” comes into play. Only then is what is left interpreted and an answer formulated for the user, via other domain-specific knowledge such as “the left-most team is given the left-most score”.

Literally, context is concerned with meaning. It describes the influence that related facts and assertions have on the meaning of a particular statement. But very often it is not clarification of meaning that we require, but an evaluation of relevance. In these cases it is not so much that context is used to fully determine the meaning of a text, but merely to determine that the meaning is not something that we are interested in — that whatever it means, it’s not relevant. This goes to the core of Lenat’s hope for Cyc that incorporating context into the huge knowledge-base will greatly reduce the processing required to answer queries and infer new knowledge. It also provides a compelling reason to consider the specifics of a task or purpose as defining the context. As SportsFinder shows, the approach of defining the knowledge units required for a task can leverage context by excluding all other knowledge. Whether, theoretically, this is by determining the meaning via context or evaluating the relevance via context seems to be a point of semantics rather than practical concern.

The techniques demonstrated by SportsFinder have been employed in a number of other successful information agents within the Intelligent Agent Lab. These agents operate in a wide variety of domains, including CASA, for searching semi-structured real-estate classified advertisements [41], Justice, which uses the context of a legal case to report summary information about judgements and rulings [98], and CiFi, an agent that retrieves citations from the World Wide Web [73]. Each agent uses lightweight purposive knowledge that both defines the target information and bounds the context of the agent’s task. Rather than dismissing these agents as ad hoc, they can be usefully viewed as exhibiting purposive knowledge as a feature.
4.6 Building a lightweight information agent

Since the original SportsFinder had been successful at reporting match results for a variety of sports and leagues, I chose, somewhat arbitrarily, to focus on the F.A. Premier League in England. I identified as a source for F.A. Premier League match results the home page of the Premier League web site\textsuperscript{16} (see Figure 4.3).

Analysing this page revealed that heuristics similar to those used in SportsFinder would be effective for this page, as described in Figure 4.4.

In order to be integrated with the Cyc knowledge-base, the new information agent, SportsReporter, needed to be implemented in SubL. The Cyc architecture has been designed to allow for the implementation of reasoning techniques other than pure logical inference. This allows Cyc to be given much more efficient ways to reason about certain types of knowledge for which logical inference would be inefficient. For instance, although Cyc could be taught to perform arithmetic by being told \texttt{(plus 1 1 2)}, \texttt{(plus 1 2 3)}, \texttt{(plus 2 2 4)} and so on, it is much more efficient to simply provide an external calculator module that can produce the appropriate result for any instance of the predicate \texttt{plus} with three arguments. These external modules are called removal modules \textsuperscript{25}, and this is the same framework that SportsReporter uses to provide Cyc with information from an external source.

The framework of removal modules that augment Cyc’s inferencing ability is quite similar in practice to that of an interface in object-oriented programming. A predicate is defined within Cyc’s knowledge-base, and a SubL function is inserted into Cyc to tell Cyc that whenever it encounters this new predicate it should not try to prove it but should pass it to an external function and wait for the result — the external function can be thought of as ‘implementing’ the predicate. Cyc will then replace the original predicate and its arguments with the result of the external function, much as described above for the example of arithmetic addition, and further reasoning can occur. In the case of SportsReporter, the predicate defined within Cyc to be implemented was called \texttt{sksiSoccerMatchResult}\textsuperscript{17}, and when called with partially unbound arguments it would use the information retrieved by the SportsReporter agent to bind its arguments to elements of the relevant match result. In this way, when Cyc encounters the predicate


\textsuperscript{17}The prefix \texttt{sksi} indicates that this predicate will be processed via a removal module using the SKSI API, although this signification is only of use to people — other internal predicates are used to tell Cyc how to handle the predicate.
Figure 4.3: Match results for the F.A. Premier League as published on the F.A. Premier League web site

Soccer match report: team name, soccer score, soccer score, team name
Team name: Arsenal, Blackburn, Chelsea, ...
Soccer score: \(0 \leq n \leq 15\)

Figure 4.4: Example heuristics used in the \textit{SportsReporter} information retrieval agent
with unbound variables, such as:

(sksiSoccerMatchResult (SoccerTeamNamedFn "ArsenalFC") ?TEAM2
 ?SCORE1 ?SCORE2)

it will call on the appropriate removal module to provide a more fully bound expression, such as:

(sksiSoccerMatchResult (SoccerTeamNamedFn "ArsenalFC")
 (SoccerTeamNamedFn "EvertonFC") 0 1)

This predicate, once filled with information retrieved from the F.A. Premier League web page, allows Cyc to reason about the match result, and combine this information with the rest of its knowledge.

SportsReporter was thus designed as a lightweight, heuristic-driven information agent, implemented in SubL, that generates a predicate that can be used in queries of Cyc’s knowledge-base. At this point, just like its predecessor, it contained hard-coded knowledge sufficient to parse a web page and identify a match report involving a particular team. These heuristics, implemented in SubL, looked like:

(define is-soccer-score? (string)
 (pwhen
   (integer-string-p string)
   (clet ((score (string-to-integer string)))
     (ret (cand (>= score 0) (< score 15))))))

which expresses whether a string from a web page or other document can be considered to signify a soccer score, and:

(define is-fapl-team-name? (string)
 (clet ((fapl-team-names-hash (make-hash-table 30 #'equal)))
 (sethash "Arsenal" fapl-team-names-hash "Arsenal")
 (sethash "Aston Villa" fapl-team-names-hash "Aston Villa")
 (sethash "A Villa" fapl-team-names-hash "Aston Villa")
 (sethash "Birmingham" fapl-team-names-hash "Birmingham")
 (sethash "Blackburn" fapl-team-names-hash "Blackburn")

---

18Strictly, the Cyc atoms such as skaSoccerMatchResult and SoccerTeamNamedFn should be prefixed with #\$ to indicate that they are defined concepts. I have omitted this prefix to aid readability.
which expresses whether a string can be considered to represent an F.A. Premier League team.

### 4.7 Integrating SportsReporter into the Cyc ontology

As with any ontology or knowledge-base, in order to be useful to Cyc, new concepts and knowledge must be attached to existing concepts and relationships, to permit reasoning on a technical level, and to make sense on an ontological and semantic level — a lone concept with no relation to any other is arguably meaningless. Several assertions were added to the Cyc knowledge-base to provide some scaffolding to help Cyc understand the \texttt{sksiSoccerMatchResult} predicate:

\begin{verbatim}
(implies (sksiSoccerMatchResult (SoccerTeamNamedFn ?TEAM1) ?TEAM2 ?SCORE1 ?SCORE2))
  (and
    (competingAgents (MatchBetweenFn ?TEAM1 ?TEAM2) ?TEAM1)
    (competingAgents (MatchBetweenFn ?TEAM1 ?TEAM2) ?TEAM2)
    (scoreInAction ?TEAM1 (MatchBetweenFn ?TEAM1 ?TEAM2) ?SCORE1)
    (scoreInAction ?TEAM2 (MatchBetweenFn ?TEAM1 ?TEAM2) ?SCORE2)))

(implies
  (and
    (scoreInAction ?TEAM1 ?MATCH ?SCORE1)
    (scoreInAction ?TEAM2 ?MATCH ?SCORE2)
    (greaterThan ?SCORE1 ?SCORE2))
  (winnerInConflict ?MATCH ?TEAM1))
\end{verbatim}

Here, \texttt{SoccerTeamNamedFn} is a function denoting a non-atomic term. It is important to note that it was not necessary to define the predicates \texttt{competingAgents}, \texttt{scoreInAction}, \texttt{greaterThan} and \texttt{winnerInConflict}, as these general predicates were already present in the Cyc knowledge-base.

As can be seen in Figure 4.5, the predicate \texttt{winnerInConflict} can be applied to any pair of individuals in the Cyc knowledge-base that are an instance of a \texttt{ConflictEvent}
Predicate: \textcolor{blue}{\texttt{winnerInConflict}}

GAF Arg: 1

Mt: \textcolor{blue}{\texttt{UniversalVocabularyMt}}
\texttt{isa}: \texttt{\langle ActorSlot \rangle}

Mt: \textcolor{blue}{\texttt{BookkeepingMt}}
\texttt{quotedisa}: \texttt{\langle Has Been Reviewed in Role Predicate Sweep \rangle}

Mt: \textcolor{blue}{\texttt{UniversalVocabularyMt}}
\texttt{genIPreds}: \texttt{\langle successfulForAgents \rangle}
\texttt{arity}: \texttt{\langle 2 \rangle}
\texttt{arg1isa}: \texttt{\langle ConflictEvent \rangle}
\texttt{arg2isa}: \texttt{\langle Agent-Generic \rangle}

Mt: \textcolor{blue}{\texttt{AgentGMt}}
\texttt{arg2Format}: \texttt{\langle Set The Format \rangle}
\texttt{(argFormat winnerInConflict 2 SetTheFormat)}

Mt: \textcolor{blue}{\texttt{UniversalVocabularyMt}}
\texttt{(arg1isa winnerInConflict 2 Agent-Generic)}
\texttt{(arg1isa winnerInConflict 1 ConflictEvent)}
\texttt{comment}: \texttt{\langle (winnerInConflict CONFLICT VICTOR) means in the ConflictEvent CONFLICT, the \#Agent VICTOR was the winner. \rangle}

\textbf{Figure 4.5:} Predicate \texttt{winnerInConflict} as defined in Cyc
**TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS**

**Collection :** SoccerTeam

**GAF Arg :** 1

- **Mt : UniversalVocabularyMt**
  - isa: ExistingObjectType SportsTeamTypeBySport

- **Mt : IRExpansionRulesMt**
  - isa: QAClarifyingCollectionType

- **Mt : UniversalVocabularyMt**
  - gens: "The collection of organizations of athletes who play soccer."

- **Mt : SportsMt**
  - conceptuallyRelated: SoccerPlayer Soccer

- **Mt : BaseKB**
  - definingMt: SportsMt
  - focalTermTypeForInducedTemplateType: (FormulaTemplateInductionTopicTypeFn SoccerTeam)

- **Mt : UniversalVocabularyMt**
  - rewriteOf: (SubcollectionOfWithRelationToFn SportsTeam focalActivityType (PlayingFn Soccer))

- **Mt : FrenchLexicalMt**
  - termStrings: "équipe de foot" "équipe de football"

**Figure 4.6: SoccerTeam concept as defined in Cyc**

and an instance of an Agent-Generic, or any of their descendants. By telling Cyc that the result of the denoting function SoccerTeamNamedFn is a SoccerTeam:

(resultIsa SoccerTeamNamedFn SoccerTeam)

and then attaching the SoccerTeam collection to the appropriate place in Cyc’s hierarchy of concepts (see Figures 4.6 and 4.7), individuals of the collection SoccerTeam can now be used with the predicate winnerInConflict. Making similar arrangements for the predicate MatchBetweenFn:

(resultIsa MatchBetweenFn SportsMatch)

means that the knowledge newly added to Cyc about the result of last weekend’s F.A. Premier League match involving Arsenal can now be reasoned about via all the relevant pre-existing concepts and relationships in Cyc. This is a significant result because the ontological engineer who originally defined the winnerInConflict predicate
Figure 4.7: Constant SoccerTeam in Cyc hierarchy of concepts
did not have to foresee that it would be applied years later to a sporting match, let alone one sourced dynamically from a web page\textsuperscript{19}.

By crafting the scaffolding by which the SportsReporter agent is integrated with the Cyc knowledge-base carefully, unforeseen yet ontologically valid inferences and conclusions can be made based on the small pieces of information provided by SportsReporter. For instance, another ontological engineer working on a completely separate area of Cyc’s huge knowledge-base — say, human emotional reactions to events — might have used the winnerInConflict predicate to assert that:

\[
\text{(implies (winnerInConflict ?PERSON) (happy ?PERSON))}
\]

and now, with no extra effort, Cyc can infer that if Arsenal defeated Everton then they should be happy about it. With a little more work, Cyc could be told that any fans of Arsenal should also be happy, and without too much imagination it is possible to see various distantly-related social and economic consequences being predicted just from this simple match report. This is possible because of the vast general knowledge already present in Cyc.

\section{4.8 Moving task-defining heuristic knowledge into Cyc}

Because the knowledge used by SportsReporter to identify and interpret match results is at this point hard-coded into the agent, any changes require source-level modifications to be made. I claimed earlier that the design of SportsReporter’s ancestors makes them both flexible in terms of applying them to other domains without modification, and also simple in that when their heuristics do need to be modified, adjusting their behaviour does not require any architectural or design consideration. However, there are degrees of simplicity, and modifying a program’s source is certainly less than ideal. As I described earlier in this chapter, I see the future of this work as being a Cyc that is able to put together the information necessary to generate lightweight information agents on demand, providing them with the details required to find the answer to whichever question Cyc currently requires be answered.

Therefore, the next step of integrating the SportsReporter information agent with Cyc was to move the knowledge encoded within it into the Cyc knowledge-base. This made the SportsReporter agent more pure in a design sense, but was also a move toward

\textsuperscript{19}Indeed, the Internet as we know it today barely existed when the Cyc project began.
providing Cyc with the means to generate new versions of the SportsReporter agent as needed to fulfill its information needs. This was not done to make the SportsReporter agent more elegant or lighter in its implementation, but to reduce Cyc’s dependence on the heuristics hand-coded into the agent. Translating these heuristics into Cyc’s knowledge-base not enables Cyc to modify them as necessary to answer different questions, but also makes them available to other users of Cyc to co-opt for purposes I haven’t imagined.

Constants were created within Cyc to represent the different concepts present in SportsReporter’s heuristics. The definition of a valid soccer score became a unary predicate stating that:

\[(\text{scoreInSoccerMatch} \ (\text{IntegerFromToFn} \ 0 \ 15))\]

Rather than simple strings for team names, the \texttt{SoccerTeam} concept was used as a collection, with individuals created for each team currently playing in the F.A. Premier League, such as \texttt{ArsenalFC} and \texttt{ManchesterUnitedFC}. Because not all soccer teams play in the F.A. Premier League, I initially used a new class of soccer team called \texttt{FAPremierLeagueTeam}, and told Cyc:

\[(\text{isa} \ \texttt{SoccerTeam} \ \texttt{FAPremierLeagueTeam})\]

However, using inheritance in this way is somewhat heavy-handed and could cause complications later (for instance when teams are relegated and promoted at the end of each season — this shouldn’t change the nature of the team). Ontologically, inheritance is a fairly restricting relationship and I opted instead to express membership in a league by a relation rather than as an existential property, so that a team is defined and described thus:

\[(\text{isa} \ \texttt{ArsenalFC} \ \texttt{SoccerTeam})\]
\[(\text{nameString} \ \texttt{ArsenalFC} \ "\text{Arsenal FC}" )\]
\[(\text{nameString} \ \texttt{ArsenalFC} \ "\text{Arsenal}" )\]
\[(\text{playsInLeague} \ \texttt{ArsenalFC} \ \texttt{FAPremierLeague})\]

Now that Cyc knows what a valid soccer score is and what teams play in the F.A. Premier League, it is possible to replace the hard-coded functions in the SportsReporter agent that contained these pieces of knowledge. Where the SportsReporter agent previously had the function:
(define is-soccer-score? (string)
  (pwhen
    (integer-string-p string)
    (clet ((score (string-to-integer string)))
      (ret (cand (>= score 0) (< score 15))))))

it can now call on Cyc to make this judgement:

(define is-soccer-score? (string)
  (pwhen
    (integer-string-p string)
    (clet ((score (string-to-integer string)))
      (pif (new-cyc-query
             '#$admittedSentence (#$scoreInSoccerMatch ,score) #$SportsMt)
           (ret T)
           (ret NIL))))))

The admittedSentence predicate is a way to ask Cyc if a fully-bound clause is acceptable given what Cyc knows. In this case, having already told Cyc that scoreInSoccerMatch (IntegerFromToFn 0 15), the SportsReporter agent can then use the admittedSentence predicate to check the legitimacy of statements of the form scoreInSoccerMatch x — if x can be unified with (IntegerFromToFn 0 15) then Cyc will reply positively, and otherwise the reply will be negative.

Similarly, where knowledge about team names had previously been encoded into the SportReporter agent’s source:

(define is-fapl-team-name? (string)
  (clet ((fapl-team-names-hash (make-hash-table 30 #'equal)))
    (sethash "Arsenal" fapl-team-names-hash "Arsenal")
    (sethash "Arsenal FC" fapl-team-names-hash "Arsenal")
    (sethash "Aston Villa" fapl-team-names-hash "Aston Villa")
    ...

the agent can now access all of this knowledge via a query to Cyc:

(define is-team-name? (string)
  (pif (new-cyc-query '#$and ($isa ?TEAM #$FAPremierLeagueTeam) ($nameString ?TEAM ,string)) #$SportsMt)
(ret string)
(ret nil))

Thus around 50 repetitive lines of the SportsReporter agent’s source code have been replaced with a single function call, but that is only an incidental benefit. This is a more extensible and re-usable approach, not only in the sense of programming and source code but more significantly in terms of functionality and meaning. Since the task-specific knowledge has been moved into Cyc, the SportsReporter agent requires fewer modifications in order to be applied to a new league or even a new sport; correspondingly, Cyc now knows about some English soccer teams and in which league they play, information that is now available for use in ways that I might not have considered, but that may be valuable in the future when they become the context to someone else’s task.

Another piece of knowledge hard-coded into SportsReporter was the location of the web page that contains the match results that are wanted. This knowledge can also be moved into Cyc, creating a binary predicate called `matchResultsWebPageURL` that relates a web page to a sports league:

```
(matchResultsWebPageURL FAPremierLeague
    "http://www.premierleague.com/fapl.rac?command=forwardOnly&
    nextPage=homepage&tab=results")
```

This allows the replacement of another hard-coded constant in the SportsReporter agent, but it also makes it easy to change which web page is used to retrieve match results for the F.A. Premier League. As the concept of a league has already been defined within Cyc’s knowledge-base via the `FAPremierLeague` constant, another level of reasoning can be added here. Because a team identifier was provided when SportsReporter was originally invoked, it can query Cyc to find out which league that team plays in:

```
(clet (league
    (first
        (query-variable '?LEAGUE
            `(#$playsInLeague ,team ?LEAGUE) #$SportsMt))))
```

and then via another query it can find the URL for a web page that reports match results for that league:

```
(clet (results-page-url
```
(first
  (query-variable '?URL '($matchResultsWebPageURL ,league ?URL)
   #$SportsMt)))

The query-variable function is very similar to the new-cyc-query function, but instead of simply returning true or false it returns a list of values that Cyc was able to unify with the specified variable, in this case ?URL. The first result is taken for now, although the possibility exists to accept more than one.

4.9 Generalising the SportsReporter heuristics

One significant piece of knowledge still encoded explicitly in the SportsReporter agent is how to actually identify a match report. I began with a heuristic adapted from the previous incarnations of SportsReporter:

Score format:  team name ... score ... score ... team name

A strong motivation for moving knowledge from SportsReporter into Cyc was to pave the way for the possibility of generating similar agents on demand. Although the heuristic for soccer match reporting given above has proved to be widely applicable, other formats are also used, one common example being:

Score format:  team name ... score ... team name ... score

which leads to the timeless driving joke of, when passing a sign that shows the distance to the upcoming suburbs, such as Moorabbin 3, Mordialloc 5, calling out “That’s a good win for Mordialloc!”.

Since Cyc has now been told about leagues that teams play in and web sites that report match results for leagues, it is appropriate to add information about how the elements of the match report might appear. Up until now, I had expressed this in SportsReporter’s source code, by looking explicitly first for a team name and then a score, followed by a second score and then a second team name. To move this knowledge into Cyc, I created a MatchReportFormat collection as a specialisation of a List, from which I extended a SoccerMatchReportFormat collection. I then created an individual member of that collection called FAPremierLeagueMatchReportFormat. As a List, the type of each element of FAPremierLeagueMatchReportFormat can be defined as follows:
(tupleMemberIndex FAPremierLeagueMatchReportFormat SoccerTeam 1).
(tupleMemberIndex FAPremierLeagueMatchReportFormat scoreInSoccerMatch 2).
(tupleMemberIndex FAPremierLeagueMatchReportFormat scoreInSoccerMatch 3).
(tupleMemberIndex FAPremierLeagueMatchReportFormat SoccerTeam 4).

This simply says that the first element of an FAPremierLeagueMatchReportFormat is a SoccerTeam, the second element is a scoreInSoccerMatch, and so on. I replaced the parts of SportsReporter that look for a match report in the contents of a web page with a general loop that adjusts to each of the elements of a provided MatchReportFormat, which has been retrieved from Cyc via a function call similar to those already introduced:

(clet (cyc-report-format-constant
  (first
   (query-variable '?FORMAT
    ('(#$matchReportFormatForLeague ,league ?FORMAT)
     #$SportsMt)))))

This only gives the Cyc constant that represents the match report format that needs to be used, not the elements of the format, which can be retrieved similarly:

(clet (next-report-element-type
  (first
   (query-variable '?X '(#$tupleMemberIndex
    ,cyc-report-format-constant ?X
    ,curr-pos-in-report-format) #$SportsMt])))

Having thus obtained the Cyc constant representing the type of the next element of the match report to look for, SportsReporter can then search through the web page and compare its elements. Because the ‘type’ of the match report element is a Cyc constant, the only way to compare anything to it is to ask Cyc again, so a final query is made, either

(new-cyc-query
  '(#$admittedSentence (,next-report-element-type ,web-page-element))
  #$SportsMt)

if the web page element is an integer, or
(new-cyc-query
   '($and ($isa ?X ,next-report-element-type)
       ($nameString ?X ,web-page-element))
   #$SportsMt)

if it is a string. The latter query involves two clauses because the string used to
represent a Cyc constant in natural language is usually different the the name given to
the constant; for example, the word “Arsenal” is commonly used to signify the Arsenal
Football Club, but ArsenalFC is a more appropriate name for the Cyc constant that
represents the club, since Cyc might need to know about other things in the world
that are also identified by the word “arsenal”. The Cyc predicate nameString was used
earlier to attach strings to Cyc constants; these are words or phrases that are known to
be used to signify the constant in natural language expressions. A constant can have
many such name strings. The query above simply asks if Cyc knows of a thing that
is of the correct type and can be written as the current web page element in natural
language.

With this last set of queries, almost all of the heuristic knowledge encoded in
SportsReporter has now been transferred into Cyc, expressed in CycL assertions and
relationships that can be used by both the SportsReporter agent and by any other user
of Cyc, human or software.

4.10 Conclusions

I have presented work toward a framework for building and eventually generating in-
formation agents designed to fulfill specific tasks, in this case augmenting the Cyc
knowledge-base with information from external sources such as web sites. I have illus-
trated through detailed explanations and examples how a lightweight, purposive infor-
mation agent can maintain a symbiotic relationship with a huge, general knowledge-
base, and how the two can leverage each other’s capabilities to produce unforeseen
beneficial results.

I have also shown that the final work presented here is significantly more general
and applicable than previous efforts. To give a brief illustration of this, consider the
prospect of adding to Cyc knowledge of the most recent match results from the Italian
Serie A football competition. Through the SportsReporter agent, implemented as a
removal module, Cyc has all of the infrastructure required. Aside from the creation
of a concept for the league, perhaps ItalianSerieA, the first pieces of new knowledge
needed would be the URL of a web page or other resource that contains the match result (such as in Figure 4.8): 

(matchResultsWebPageURL ItalianSerieA
   "http://www.soccerstats.com/latest.asp?league=Italy")

and the format in which that page presents match results, in this case:

(tupleMemberIndex ItalianSerieAMatchReportFormat SoccerTeam 1).
(tupleMemberIndex ItalianSerieAMatchReportFormat scoreInSoccerMatch 2).
(tupleMemberIndex ItalianSerieAMatchReportFormat scoreInSoccerMatch 3).
(tupleMemberIndex ItalianSerieAMatchReportFormat SoccerTeam 4).

and so on for the other teams in the league.

At this point it might be observed that the match report format is the same as for the F.A. Premier League, and so an intermediary re-usable concept might be created, such as SoccerMatchReportFormatTeamScoreScoreTeam, although perhaps this is a little cumbersome. Regardless, Cyc would then need to be told about the teams that compete in the league:

(isa SampdoriaUC SoccerTeam)
(nameString SampdoriaUC "Sampdoria Unione Calcio")
(nameString SampdoriaUC "Sampdoria")
(playsInLeague SampdoriaUC ItalianSerieA)

At this point Cyc knows everything that is required to reason about match results from the Italian Serie A just as it can for the F.A. Premier League. When the sksiSoccerMatchResult removal module is passed a team it will ask which league that team plays. It will then use that information to query Cyc for the URL of the web page that contains match results for this league, as well as for the match report format, which it can then use to retrieve and parse the match results and return them to Cyc.

Because SportsReporter is intrinsically a web-based information retrieval agent, it would not be as simple to use it to retrieve information from a non-web source as it is to change the particular web source that it queries. However, the changes required to operate in a different medium would not be great. Modifying the underlying transfer method from HTTP to something else would be quite easy, and the only other significant effort required would be to help Cyc to understand that

Figure 4.8: Match results for Italian Serie A League as published on the Soccerstats web site
there is more than one way to find a match result, perhaps by creating a predicate like `sksiSoccerMatchResultBySOAP` and a couple of relationships to explain that `sksiSourceAccessMethod DeutscheBundesLiga SOAP`.

Of course, I hope that it has been obvious throughout this chapter that all of the content presented here is in no way limited to soccer — `sksiSoccerMatchResult` could (indeed should) easily become `sksiMatchResultByHTTP`, with the creation of a rule that explains how to win a non-soccer sports match. And a further hope is that it doesn’t take much imagination to see that what I have been describing is really a flexible framework for enabling Cyc to build a lightweight module to answer any particular question that requires external information.

An ongoing effort within the Cyc development team is the processing of natural language expressions to populate Cyc with new knowledge — effectively providing Cyc with the ability to ‘read’ the World Wide Web\(^\text{21}\). The framework I am describing is in a way the opposite of this effort: augmenting Cyc by providing the means to construct and deploy purposive information agents to interoperate with external systems and information sources, allowing it to draw upon the oceans of data that is being interconnected as the Internet continues to reach into previously private or inaccessible data stores.

As this data is integrated with the Cyc ontology, it becomes semantically richer — to the extent that Cyc might serve as a lingua franca for interoperating information systems, served by lightweight, purposive information agents.

\(^{21}\text{Based on personal communication with Cyc developers.}\)
Chapter 5

Automatic Reconciliation of XML Structures

5.1 The EBFM Algorithm

In 1998, an algorithm for mapping between elements in data sources was presented by Itoh, Ueda and Ikeda [56]. Extending and adapting this algorithm has been a core part of the practical side of work conducted toward semantic interoperation in the Intelligent Agent Lab\(^1\) for the last decade or so. The algorithm, called Example-Based Frame Mapping (EBFM) by [56], finds the representations of common entities in different databases or XML documents, and from pairs of such representations deduces correspondences between the fields or schemas of the databases or XML documents, respectively. By comparing the instances of fields from two information sources, fields that share a significant number of instances are considered to be semantically related, an idea similar to that presented in 2005 by Avesani \textit{et alia}, who wrote of comparing hierarchical taxonomies of document categories: the meaning of two nodes is equivalent if the sets of documents classified under those nodes have a meaningful overlap [8].

AReXS (Automatic Reconciliation of XML Structures) is an application that reconciles heterogeneous information sources by aligning them according to their implicit ontological structure [54, 66, 125, 67, 68, 69, 70]. AReXS\(^2\) reconciles differences of expression and representation between heterogeneous data sources without any predefined

---

\(^1\)The Intelligent Agent Lab is part of the Department of Computer Science and Software Engineering at The University of Melbourne — http://www.csse.unimelb.edu.au/agentlab

\(^2\)The name AReXS was coined by Dominic Hou, who wrote an implementation of the original EBFM algorithm for his Master’s thesis in the Intelligent Agent Lab at The University of Melbourne. Lacking imagination, and to recognise his work, the AReXS name was kept as the application described in this chapter was developed.
knowledge or human intervention. It achieves this by identifying XML elements whose meanings are similar enough to be considered equivalent. AReXS requires no knowledge or experience of the domain in which it works, and thus is domain-independent. By requiring no domain knowledge, AReXS is suitable for application to any field; its success relies on its ability to identify and resolve the differences in representation that result from sourcing data from a heterogeneous environment. The EBFM algorithm, and thus AReXS, closely mimics the intuitive process that a person is likely to follow when tasked with aligning multiple structured information sources, as I’ll explain in this section.

A notable characteristic of the algorithm is that it ignores field or tag names, considering only their instances. This is not because no value can be gained from these aspects of information sources – as described in Chapter 3, much effort has been invested in approaches to matching schema and aligning ontologies by the names of fields and concepts – but because in practice such names and labels are often not as readily available and as informative as is needed for such methods to succeed by themselves (for instance, Drumm et alia observe that “legacy systems are mostly not optimized towards fostering system interoperability. The schemas typically make extensive use of technical names, abbreviations, and proprietary structure” [30]). Neither approach, in isolation, seems likely to solve the problem of semantic interoperability; rather, strategies that combine several approaches and make use of all the information available in a given situation will surely be the most effective in the end. However, it is very often necessary in science to work on a single aspect of a problem in isolation to make advances that only later are recombined to advance the state of the art.

As a demonstrating example taken from my experience with real data, consider two XML documents (or databases, or any other structured data sources) that contain information about cars. Data representations of real-world objects, from database tables and XML documents to formal ontologies and programming language classes, capture descriptions of characteristics of the objects that they seek to represent. Therefore, since a defining characteristic of a car is its model, any meaningful representation of a car will include a representation of its model. In one document in this example, the representation of each car may include an element called ‘make_and_model’, and so there may be a record that contains <make_and_model>Mitsubishi Verada</make_and_model>. In the other document, however, the representation of each car may use an element called ‘description’ to encode the same information, and so contain <description>Mitsubishi Verada</description>. The difference in representation will hinder any attempt to integrate these two data sources, and will prevent the systems that contain and maintain
them from interoperating.

The representations could diverge further in how they encode other characteristics of their objects; for instance, the first representation may use ‘price’ where the second uses ‘cost’ to express the value of a car, and so on. The general approach to solving problems of this nature is to reconcile the ontologies of the two data sources (or the systems which maintain them), and a number of methods have been proposed. However, as discussed in Section 3, these methods are typically either linguistic or graphic; linguistic methods tend to use thesauri to find synonyms among field names or element tags, while graphic methods look for similarities in the hierarchical relationships between field names and tags. Apart from the fact that neither of these approaches has so far led to a complete solution, each approach relies on certain assumptions that are not always valid. Field or element names chosen in practice by system designers and developers cannot be relied on to form neat synonym pairs, partly due to the fact that natural language is capable of capturing meaning in such a variety of ways. As mentioned, in the data gathered for the experiments presented in this thesis, the terms ‘make_and_model’ and ‘description’ have been used to represent the same information, and it seems unlikely that any thesaurus would identify these as synonymous.

On the other hand, data sources often do not come with a well-formed ontology from which hierarchical and other structural information can be extracted in order to apply graphic methods. This seems particularly to be the case when trying to integrate data that has been published by independent sources, since even though a formal ontology or schema may have been defined for the development of the system that generated the data, any such ontology is typically not part of the published content.

Further, I have found that often the data to be integrated, while being presented in such a form that fields or elements are readily discernible, does not have easily identifiable labels or names for the fields, which obviously makes it very difficult for either of these two methods to produce useful results.

Finally, implementations of these methods are typically only semi-automatic, and it is desirable to aim for as much automation as possible in order to achieve scalable solutions.

Rather than analysing the field or element names present in XML documents or data sources, the EBFM algorithm on which the AReXS application is based looks to the data instances within the sources to infer semantic correspondences between their fields. Assuming that these two data sources of information about cars contain information on popular or common cars, it is likely that there will be cars that are represented in both data sources. Finding the pair of data instances that represent
these cars in each data source will generally be possible despite the data sources having different or missing field names, because the actual field contents for instances of the same car should be similar.

To return to the original example, if two data sources contain instances with fields `<make_and_model>Mitsubishi Verada</make_and_model>` and `<description>Mitsubishi Verada</description>` respectively, this suggests that the fields labelled `make_and_model` and `description` are in some way semantically equivalent, based on the similarity between their contents. If there are similar matches between the other fields of these two instances then the suggestion becomes stronger, and so this pair of instances will be useful in reconciling the schemas of the two data sources. The more pairs of instances with matching fields can be found, the more confidence there can be of the semantic equivalence of the fields which forms the basis of both the original EBFM algorithm and my own work.

However, the assumption that the same real-world objects will be described in each data source is not applicable to all domains, and it is on these domains that I have focused my attention. Unlike product catalogues, which contain descriptions of products rather than actual items, I am interested in domains such as classified advertisements for rental properties or second-hand vehicles, where each item described is a particular real-world object. The key difference is that each shop that stocks a particular area of products, such as laser printers, is likely to stock the same item, perhaps a ‘LaserPrinter 2000’, and so it will be possible to find the representation of this product in each shop’s catalogue, and thus use it to infer the appropriate mapping between fields. For second-hand vehicles, on the other hand, it seems to be uncommon for any individual seller to advertise their vehicle in more than one classified listing service, presumably due to the cost of doing so.

Because of this, the assumption that a particular vehicle will be represented in each data source is not valid and so this cannot be relied upon when searching for pairs of instances to match fields. To overcome this, the basic principle from the original algorithm is retained but each field is considered independently of the schema of which it is a part. Instead of searching for pairs of instances with many matching fields, the AReXS implementation of EBFM seeks only pairs of fields with similar contents. This relaxation weakens the algorithm somewhat in domains where each data source contains representations of the same objects, but aids the matching of fields from data sources where similarity of instances cannot be assumed.
5.2 Extending the EBFM algorithm

AReXS has been implemented to process pairs of simple XML documents, as this is a convenient data format that lies near the middle of the spectrum of data formats that ranges from highly structured relational databases at one end to loosely structured textual documents at the other end. The transformations necessary to convert data in any form in this spectrum to an XML document are readily imaginable (SQL, XSLT, wrappers, scrapers, et cetera). An XML document can easily be read as containing instances, represented by second-level tags, with fields, represented by third-level tags; the data within the tags is the contents of the fields.

The algorithm implemented by AReXS consists of three stages: transforming a pair of XML documents into a ‘frames and slots’ data structure, identifying potential matches between slots from each data source, and calculating a confidence score for each potential match.

Transforming XML documents into frames, slots and instances

The first stage of the algorithm implemented in AReXS consists of reading a pair of XML documents and storing their contents in two internal data structures. The data structures closely resemble the original XML documents, being a set of slots that comprise a frame in the context of the XML document and a set of instances of that frame.

For example, given the following XML-like document:

```xml
<Autotrader>
  <car>
    <make model>Toyota Corolla SECA</make model>
    <yr>1993</yr>
    <kms>140,000</kms>
    <price>$9,995</price>
    <state>WA</state>
    <media>?</media>
  </car>
  <car>
    <make model>Ford Falcon</make model>
  </car>
</Autotrader>
```

AReXS will only process XML documents that follow this structure, so it is not generally applicable to all XML documents.

---

\(^3\)AReXS will only process XML documents that follow this structure, so it is not generally applicable to all XML documents.
the top-level tags identify the data source as ‘Autotrader’, the second-level tags identify instances of the data object for this data source, in this case labeled ‘car’, and the third-level tags identify the slots and slot values that comprise this instance. In other words, for this data source the frame consists of the following set of slots:

make_model, yr, kms, price, state, media, dealer

and there are two instances, whose slots have the values:


and


Thus individual slot values can be referred to as the value of the $i$th slot in the $j$th instance of the $k$th data source, although I’ll stick to ordinals such as first, second, third wherever possible.

A second XML-like document might define a data source thus:

<AutoWeb>
  <vehicle>
    <year>1990</year>
    <description>BMW 3 18I</description>
    <price>$8,500</price>
    <body_type>Coupe</body_type>
    <colour>red</colour>
    <location>Melbourne</location>
  </vehicle>
</AutoWeb>
and would likewise be represented by a frame of slots:

\[
\text{year, description, price, body\_type, colour, location}
\]

with two instances. The value, therefore, of the third slot ('price') in the second instance of this second data source would be ‘$5,990’.

Frames can be labelled and referred to by the name of the second-level tag in their XML document, e.g. ‘car’ and ‘vehicle’ in this example, but this information is not used by the algorithm.

**Identifying potential matches between slots from each data source**

To identify potentially matching pairs of slots, a filtering function is applied to all pairs of slots in turn, e.g. \{make\_model, year\}, \{make\_model, description\}, \{make\_model, price\}, et cetera. This filtering function uses a comparison function to evaluate the similarity of two slot values.

\[
\text{ff(slot, slot)} = \begin{cases} 
1, & \text{if there exists an instance pair such that} \\
& \text{cf(slot value, slot value)} > \text{cf threshold} \\
0, & \text{otherwise} 
\end{cases}
\]

The pairs of slots that pass this filtering stage form a set of candidate matching field pairs for the third stage of the algorithm. Along the way, those instances pairs for which \text{cf} is sufficiently high for a given pair of slots are retained as supporting evidence for that hypothesis.

The comparison function, although necessary to the algorithm, is somewhat arbitrary, in that any function that takes as input two slot values and generates a value that represents the similarity of the two slot values can be used, and in this way, it is the comparison function that defines the term similar in the context of this algorithm. In AReXS, an \text{n}-gram based string comparison function has been used, similar in nature...
to the Character-Based Best Match algorithm introduced by [105]. The implications of this will be discussed later.

For the pair of example data sources described earlier, the filtering function would first be applied to the pair of slots \{make, model, year\}. The comparison function would be calculated for the values of these slots in the first instances of each data source, in this case ‘Toyota Corolla SECA’ and ‘1990’. If the result exceeds a preset threshold (discussed later), this instance pair will be recorded as supporting a match between the slots ‘make\_model’ and ‘year’, and then the value of the ‘make\_model’ slot in the first instance of the first data source will be compared to the value of the ‘year’ slot in the second instance of the second data source, ‘1984’. If the result of the comparison function again exceeds the predetermined threshold, this pair of slots will also be recorded as supporting a match between this pair of slots.

So the first stage of the algorithm produces a filtered set of candidate slot pairs, and for each such pair a set of supporting instance pairs and the corresponding similarity score for their slot values.

Calculating a confidence score for each potential match

For each candidate slot pair, a confidence score is calculated based on the slot pair’s supporting instance pairs. This confidence score is derived by calculating a group of three functions for each instance pair that supports the slot pair, and then combining the results of these three functions across the set of supporting instance pairs. The three functions are taken from the original EBFM algorithm as described by Itoh et alia. To aid in reading the function definitions that will be introduced in the following pages, it is necessary to define some terms:

Let $Z_1$ and $Z_2$ be the sets of instances from the first and second information sources respectively.

Let $z$ be an instance from an information source.

Let $zp$ be a pair of instances consisting of $z_1$ and $z_2$ being instances from the first and second information sources respectively, and let $ZP$ be a set of pairs of instances.

Let $s$ be a slot from an information source.

Let $sp$ be a pair of slots consisting of $s_1$ and $s_2$ being slots from the first and second information source respectively that comprise the slot pair.
Let $\text{val}(z, s)$ be the value of a slot $s$ in an instance $z$, and let $v_1$ and $v_2$ be the values of a pair of slots for a pair of instances.

The first function is the comparison function $\text{cf}$ already used to express the similarity of the values of two slots.

The second function is a singularity function $\text{sf}$ that rewards a slot pair if, for a given supporting instance pair, the slot values for that instance pair do not match well with slot values from other pairs of instances from the information sources. So where the first stage of the algorithm identified pairs of slots for which there were strong supporting instance pairs, this second stage now checks that the slot values of these supporting instance pairs don’t match values from other instances as well. This function is defined thus:

$$\text{sf}(sp, zp) = \frac{2 \text{cf}(v_1, v_2)}{\sum_{z \in Z_2} \text{cf}(v_1, \text{val}(z, s_2)) \sum_{z \in Z_1} \text{cf}(v_2, \text{val}(z, s_1))}$$

The third function is a weighting function that calculates the uniqueness of the slot values of a supporting instance pair among the whole set of supporting instance pairs, and, following Itoh et alia, is calculated thus:

$$\text{wf}(zp, sp) = C \cdot \text{uniq}(\text{first slot, supp inst}_1) \cdot \text{uniq}(\text{second slot, supp inst}_2)$$

where $C$ is a constant that can be used to tune the algorithm (I have found 0.5 to be effective on the data sets I will discuss later) and $\text{supp inst}_1$ and $\text{supp inst}_2$ are the sets of instances from the two information sources respectively that support the given proposed matching slot pair.

The function $\text{uniq}$ is derived from information theory and is a calculation of the probability of a slot value given the other values of that slot in a set of instances. If $\text{uniq}$ takes as input a slot and a set of instances, and $n$ is the number of instances in that set, then

$$\text{uniq}(s, Z) = \frac{1}{n \log n} \sum_{j=1}^{n} \log \frac{n}{c_j}$$

Itoh et alia give a detailed derivation of this function from first principles, but it will suffice to say here that the result will range between 0 and 1, with 0 indicating that all the values of the chosen slot in the set of instances are the same and 1 indicating that all the values are different. In other words, a result of 1 means that the slot values
are highly unique whereas a result of 0 means that they are identical. This calculation helps to determine the confidence that can be placed in any proposed matching pair of slots.

Having defined these three functions, they are applied, for a given potentially matching slot pair, to the set of supporting instance pairs for that slot pair as follows:

\[
\text{confidence}(sp, ZP) = 1 - \prod_{zp \in ZP} 1 - (cf(v_1, v_2) \cdot sf(sp, zp) \cdot wf(zp, sp))
\]

This produces a final confidence score for a potentially matching slot pair, ranging from 1 as the highest confidence to 0 as the lowest possible score.

5.3 On the similarity of data instances

One of the fundamental elements of the instance-based frame mapping algorithm presented in the previous section is the notion of similarity; although this is not the primary focus of this chapter, it is nonetheless an interesting topic worthy of some consideration. The comparison function \( cf \) directly compares two elements of data, or symbols, and computes a similarity score, which is then interpreted as a measure of the semantic distance between the two symbols. The definition of this comparison function has, undoubtedly conspicuously, been omitted; this is because, in technical terms, semantic distance is a vague concept, open to debate and discussion. This is particularly complicated because, intuitively, judgements of similarity seem to be quite easy to make, yet when specificity is required, people often find themselves at a loss to draw lines and make definitive statements about the nature of things. This is a problem that has plagued ontology since its earliest days, for as soon as two concepts are identified, the question arises as to how they are related, and the most basic of relations is sameness or difference. My first philosophy teacher asked me if half an apple is an apple, my first linguistics teacher asked me if a cup without a handle is still a cup, and my first doctoral supervisor asked me the difference between a class, a subject and a course. Fortunately, as this is not a thesis of philosophy, Plato can be left in his cave and Descartes can be left to his doubts. Computer scientists do things differently: they define functions, which is exactly what must be done in order for the AReXS algorithm to work.

Pieces of data can be compared in many ways. When performed by a computer, the most basic comparison is binary identity — are the bits that define this piece of data
identical in value\(^4\) to the bits that define this other piece of data? In general, computers handle equality exceedingly well, but one of the central themes of the earlier chapters of this thesis is that real-world data is imprecise and less than perfectly structured and defined, and therefore a measure of identity is not sufficient.

Nor, for the same reasons, is a measure of equality, as might be applied to a pair of integers, or indeed any data instances. Most data structures in computer programs include some notion of equality, being an indication that two instances of data will produce the same outcome when particular operators or functions are applied. This also is insufficient for a system such as AReXS, where although it is software and thus all data is, at root, binary, the data has been created by people and is intended to be interpreted by people. Rather than equality, it is preferable to have a comparison function that can assess a range of similarity.

Data as strings

Because the majority of the data that was collected from a variety of real-world information sources was textual in nature\(^5\), it was decided to consider data instances as strings, or arrays or vectors of characters. There are several commonly-used techniques for measuring the similarity of strings, each with different characteristics, but all having in common that they measure degrees of similarity, rather than the binary possibilities of identical or not:

- Substring checking
- Edit distance calculation
- Phonetic comparison
- Thesaurus or dictionary searching
- \(N\)-gram matching

Substring checking

Substring checking involves searching for a string within another string; given a pair of strings, if either is found in the other, the strings could be said to be similar. If both strings can be found in the other, then they are identical. This gives a measure

\(^4\)High and low voltage, if one considers the operation performed by hardware, or 0 and 1 for software.

\(^5\)Images were recorded, such as photographs of cars offered for sale, but including such data in the comparison was beyond the scope of this thesis.
TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS

of similarity that, although crude, has three degrees instead of just the two degrees of identical/not identical provided by an equality or identity test.

**Edit distance calculation**

One such algorithm is the edit distance calculation. The edit distance between two strings is a measure of how many changes have to be made to one string to obtain the other string. Generally, the permissible changes are deleting a character from the string and inserting a new character somewhere in the string. Variations on the algorithm allow transposing of characters and other less common changes. Given a string, there are many paths of changes that will lead to another particular string, so the edit distance is defined as the shortest of those paths, i.e. the minimum number of permissible changes required to turn one string into another. This would normally be implemented via linear programming.

**Phonetic comparison**

A widely-used application of string comparison is in checking spelling in word processing software. Although edit distance calculations are typically used to compare words entered by the user to the dictionary of known correct spellings, this can leave too many candidates, and so figuring out how the words might sound when spoken can significantly refine the matching process. Phonetic comparison involves translating a string of characters into a string of phonemes, or base elements of spoken language; thus, while edit distance calculations might suggest three equally close candidates for a particular misspelled word, one of those candidates might sound much more similar to the word entered by the user, and so would be a better suggest for correction.

**Thesaurus or dictionary searching**

If the strings to be compared are assumed to be words from a language, a thesaurus of synonyms can be used to look up previously determined similarities between them. For example, the strings ‘crimson’ and ‘scarlet’ might have been recorded as being similar to the string ‘red’. A function of this kind is provided by the WordNet API and has been used in several ontology alignment systems, as discussed in Chapter 3.

**N-gram matching**

An alternative to considering the individual characters that comprise a string is an n-gram representation; that is, a vector that consists, for a given string, of all its substrings
of length $n$. For example, a 3-gram or tri-gram representation of the word ‘rather’ would be the vector \{rat, ath, the, her\}, consisting of four $n$-grams, each occurring once. A count is then made of the number of elements from one vector that occur in the other, with multiple occurrences counting multiple times. The number of matches counted is then normalised by dividing by the total number of possible matches. Thus, if the string ‘rather’ was compared to the string ‘there’, the corresponding 3-gram vectors would be \{rat, ath, the, her\} and \{the, her, ere\}; the count of elements from the first vector found in the second would therefore be two, the total number of possible matches would be four, and so the similarity score would be 0.5. Obviously such a calculation only tells half the story, which can be completed by performing the calculation again in reverse, counting the elements of the vector \{the, her, ere\} found in \{rat, ath, the, her\}; this time, the count of elements found is two and the total number of possible matches is three, giving a similarity score of 0.66. The two scores can be averaged for a final result of 0.58.

The chosen comparison function

AReXS uses a \{2,3\}-gram matching comparison function to calculate the similarity of data instances. Simple substring checking is too coarse and phonetic comparison is not appropriate because the data instances found in the information sources being studied often contain non-verbal content, such as dollar signs, that seem to be important to include when considering the similarity of the data instances, but difficult to deal with phonetically. Edit distance calculation could have been used, but seems likely not to cope as well as $n$-gram matching with certain situations, such as when two fields share a common format for part of the content but differ consistently in another part. Thesaurus searching would be helpful, but only for the subset of data instances that are simple natural language words, which analysis suggests would be few.

5.4 Comparing strings and data without domain knowledge

The $n$-gram matching comparison function performs well in many of the cases found in the information sources used in this study. Often, when prompted to describe an object such as a car with little instruction, people choose to provide different levels of detail. This is demonstrated by the fact that most of the information sources containing descriptions of second-hand cars that were studied included a field similar to ‘description’ or ‘make and model’. Generally, cars can be identified by the name of the company that manufactures them, the model name given to them by the company,
and a series designation to further differentiate variants of the model. Not all of this information is needed to identify a car, since model names are unique to manufacturers, so a ‘Mitsubishi Magna TR’ can reasonably be referred to as a ‘Magna TR’ without causing any confusion, but not as a ‘Mitsubishi TR’. A domain expert may know that the Magna is the only model by Mitsubishi to have been given the TR designation, but an ordinary person certainly won’t. On the other hand, an ordinary person looking for a second-hand car will probably not understand or care about the different series within a model, and so ‘Mitsubishi Magna’ will be precise enough.

The comparison function employed in AReXS considers ‘Mitsubishi Magna’ and ‘Mitsubishi Magna TR’ to be very similar, scoring them at 0.91, and copes equally well other examples of inclusion of small extra data. However, its performance slips when asked to compare ‘Mitsubishi Magna TR’ and ‘Magna TR’, scoring them at 0.69. The missing element in this second case is the domain knowledge that ‘Magna’ implies ‘Mitsubishi’, but since AReXS is explicitly designed to function without domain knowledge, its accuracy must deteriorate in cases where such knowledge is critical. Interestingly, because the comparison function is textual, it suffers particularly from the length of the missing word ‘Mitsubishi’ — comparing ‘M Magna TR’ to ‘Magna TR’ with the same algorithm produces a score of 0.88. This case suggests that the strings might be better compared by words rather than by bi- and tri-grams, but such an approach then fails when words are merely misspelt or otherwise altered, such as ‘Mistubishi’ or ‘Mitsu’.

The real problem seems to lie in trying to find an algorithm that works in all cases. N-gram matching was chosen because it works well in many of the cases encountered in the data under study, and it was desirable to keep the string comparison function relatively simple in order to simplify the analysis of the overall frame mapping algorithm. However, one of the findings of this experiment is that there are enough cases that the chosen comparison function doesn’t handle well that it seems clear that a combined approach is essential. The evidence for this and a discussion will be presented in the next chapter.
Chapter 6

Applying AReXS to Real, Messy Information

The AReXS algorithm was applied to a number of data sets from a variety of domains: international airport weather reports obtained via web services, stock price quotes obtained similarly and second-hand car classified advertisements extracted from Australian web sites\(^1\). These domains have different characteristics, permitting a thorough exploration of the behaviour of the modified EBFM algorithm implemented in AReXS. While some domains are populated with data crafted by experts, such as that of airport weather reporting, others are filled with content entered by non-experts, represented here by second-hand car advertisements. Some domains are intended to be consumed by professionals, such as that of stock price quotes and weather reports, while others are marketed to ordinary consumers. This variety of data was deliberately chosen in reaction to the data commonly used in the literature of database integration and semantic interoperability, which generally falls into one of two categories: highly-structured data created by professionals for professionals, or neatly-prepared, ontologically sound data created by academics for academics. The purpose of these experiments is first to confirm the sorts of results commonly reported for semantic interoperability approaches and then to go beyond these results by applying a confirmed technique to data that is somewhat more “wild”, and yet representative of a significant and growing body of information that is used on a daily basis by ordinary people.

---
\(^1\)The data for the airport weather reports and stock price quotes was collected and processed by my colleague, Dr Kuldar Taveter, to whom I am grateful. I was assisted in the processing of data for the second-hand car classified advertisements by Daniel Margulius, to whom I am also grateful.
6.1 Experimental design

The AReXS algorithm was applied to the data from each domain in a relatively simple fashion. Firstly, the data was obtained by accessing various web services and sites to retrieve sets of sample records. Where possible, a large number of sample records was collected from a large number of sources (for example, 100 records were collected from each of 5 second-hand car sale web sites).

Because the data presented on web sites was not in an XML form, it was necessary to parse the web pages to extract the data. Significant care was taken to ensure that the data retained its original structure and form. Although where field names were apparent (such as in Figure 6.9) these field names were applied to the data, this was purely for cosmetic purposes, as the AReXS algorithm does not use field names or labels.

For each pair of sources in each domain, the records for both sources were compared manually to determine the correct field matches. This is an inherently subjective activity, which is complicated by the fact that it is not always clear whether a pair of fields should be judged to be equivalent. The examples presented by the original authors of the EBFM algorithm [56] appear to be quite clear-cut, as were the data that was retrieved from the weather report and stock price quote web services, whereas the data collected from second-hand car classified advertisements was in many cases anything but clear. The problems that arose from the application of AReXS to this domain, and the solutions implemented, will be discussed in depth later.

Once the desired field matches had been identified, the data from each pair of sources in each domain was submitted to the AReXS algorithm, which reported prospective matches and a confidence or strength for each. A threshold was applied to convert this scale to a binary measure of inclusion or exclusion, and the resulting field pairs were compared to the nominated correct matches.

To evaluate the algorithm’s performance, simple precision and recall measures were calculated. These two metrics express respectively the proportion of retrieved documents that are relevant and the proportion of relevant results that were retrieved [19, 104], as shown in Figure 6.1.

As the AReXS algorithm identifies proposed matches, and correct matches can be hand-picked by an expert, or chosen by surveying a number of people, the precision and recall measures can be directly adapted by substituting proposed matches for retrieved documents and correct matches for relevant documents (Figure 6.2).

Apart from the fact that these were the two measures reported by the original
creators of the EBFM algorithm [56], a survey of other evaluative techniques did not reveal any compelling reason not to employ precision and recall. Further, precision and recall have often been used as measures of the effectiveness of ontology alignment methods [30] (see also [35, 36, 38]).

Since AReXS reports a ranged score, it would have been possible to consider the proposed matches to be ranked and thus calculate various related measures from the field of information retrieval, but this was rejected because reporting that a particular match was identified more strongly than another when both are desired doesn’t add meaning to the results. However, given that it is possible for the algorithm to report that a particular field from one information source matched more than one field from another source, the relative strengths of the matches can be used to reduce the occurrence of false positives, as will be discussed later. Thus, each of the three experiments consists of a set of information sources as input, and a set of precision and recall measures as output. Some work published in the area of ontology alignment has reported results using the F-measure, a combination of precision and recall into a single value. This is done because a one-dimensional value makes direct comparisons of results simpler than the two-dimensional precision/recall pair. However, since the purpose of this thesis is to explore the experimental results in detail, there is no reason to simplify the results.
6.2 Weather reports

As an initial verification of the AReXS algorithm, and an attempt to produce results comparable to those originally reported by Itoh et alia, the domain of weather reports was chosen for its expected consistency and cleanness of data, and because like the original assumption underlying the EBFM algorithm, it would be possible to obtain data from information sources that contained representations of the same real-world entities. This plays to a strength that AReXS inherits from EBFM. Two web services were identified that provide weather reports for international airports.

The first of these services was World Weather provided by InnerGears [55], which provides a weather report for any airport in the world registered by the International Civil Aviation Organization (ICAO) based on a specified ICAO identifier. The second service chosen was Airport Weather provided by CapeScience [16], which provides similar data and can be accessed in the same way by ICAO identifiers. It was thus possible to ensure that the two sets of reports to be reconciled by AReXS would contain the reports for the same cities. Both Web services were invoked simultaneously 29 times, and the data records of weather forecasts returned served as input for the AReXS algorithm. As an example, the data returned by each service for Port Moresby Jacksons International Airport (AYPY) is shown in Figures 6.3 and 6.4.

It seems highly likely, based on the data retrieved from each source, that both services originally gained their data from a common source. This makes these information sources an especially good example of system designers diverging in their representation of data, as each service has (presumably) repackaged the data to suit their own purposes. Seven matching pairs of fields can easily be identified as the ‘correct’ results that AReXS should achieve (see Table 6.1); the actual results from running the AReXS algorithm on the two sets of weather reports are presented in Table 6.2.

With such strong results, there is no need to apply a threshold. However, in terms of precision and recall, there are some problems.

Figure 6.3 shows the raw precision and recall values for the first experiment. These results are lower than anticipated, as the data is somewhat troublesome: fields ‘1’ and ‘2’ in the first source are simply repeated subsets of a third field, ‘5’. It seems likely that any algorithm will be confused by repeated data, but it clearly does occur in practice and so it must be handled. Applying a filter that only accepts the highest scored potential match for a given field removes this confusion and increases the scores, as shown in Table 6.4.

These figures are on par with the original results reported for the EBFM algorithm,
CHAPTER 6. APPLYING AREXS TO REAL, MESSY INFORMATION

Figure 6.3: Example record collected from World Weather

Figure 6.4: Example record collected from Airport Weather
<table>
<thead>
<tr>
<th>World Weather</th>
<th>Airport Weather</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1 Location</td>
</tr>
<tr>
<td>12</td>
<td>2 Humidity</td>
</tr>
<tr>
<td>13</td>
<td>3 Pressure</td>
</tr>
<tr>
<td>9</td>
<td>4 Sky</td>
</tr>
<tr>
<td>10</td>
<td>5 Temperature</td>
</tr>
<tr>
<td>8</td>
<td>6 Visibility</td>
</tr>
<tr>
<td>7</td>
<td>7 Wind</td>
</tr>
</tbody>
</table>

**Table 6.1:** Hand-chosen field matches for weather reporting web services

<table>
<thead>
<tr>
<th>World Weather</th>
<th>Airport Weather</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 Location</td>
<td>0.9501</td>
</tr>
<tr>
<td>2</td>
<td>1 Location</td>
<td>0.9558</td>
</tr>
<tr>
<td>5</td>
<td>1 Location</td>
<td>0.9764</td>
</tr>
<tr>
<td>7</td>
<td>7 Wind</td>
<td>0.9999</td>
</tr>
<tr>
<td>10</td>
<td>5 Temperature</td>
<td>0.9785</td>
</tr>
<tr>
<td>11</td>
<td>5 Temperature</td>
<td>0.9538</td>
</tr>
<tr>
<td>12</td>
<td>2 Humidity</td>
<td>0.9992</td>
</tr>
<tr>
<td>13</td>
<td>3 Pressure</td>
<td>0.9997</td>
</tr>
</tbody>
</table>

**Table 6.2:** AReXS field matches for weather reporting web services

Precision: $\frac{5}{8} = 62.5\%$
Recall: $\frac{5}{7} = 71\%$

**Table 6.3:** Precision and recall of AReXS field matches for weather reporting web services

Precision: $\frac{5}{5} = 100\%$
Recall: $\frac{5}{7} = 71\%$

**Table 6.4:** Precision and recall of AReXS field matches for weather reporting web services considering only strongest match per field
Table 6.5: Precision and recall of EBFM with five items represented in both information sources [56]

<table>
<thead>
<tr>
<th>n = 5</th>
<th>n = 3</th>
<th>n = 1</th>
<th>n = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>100%</td>
<td>99.4%</td>
<td>99.2%</td>
</tr>
<tr>
<td>Recall</td>
<td>68.6%</td>
<td>65.1%</td>
<td>62.1%</td>
</tr>
</tbody>
</table>

Table 6.6: Precision and recall of EBFM with varying number of items represented in both information sources [56]

on data that is similar to (perhaps slightly more complicated than) that presented in the original description of the algorithm, as can be seen in Table 6.5.

The results repeated here are the averages from the best-case scenario reported by Itoh et alia [56], being that where there are at least five common records present in the information sources to be reconciled, i.e. among the items represented in the information sources, there are at least five instances of the same real-world item being represented in both information sources (refer to the original publication by Itoh et alia to see the complete range of results). This was deliberately the case for the data sets retrieved from the weather reporting web services. With a lower number of items present in both information sources, although the precision stays very close to perfect, the recall for EBFM decreases significantly (see Figure 6.6).

As shown in Figures 6.3 and 6.4, the data records returned by the weather reporting web services do not always contain meaningful tags identifying fields. None of the many ontology reconciliation algorithms that depend on the presence of meaningful field or concept labels could therefore manage to reconcile such information sources. The AReXS algorithm is able to perform the reconciliation: the elements of the data records returned by the first Web service can be mapped to the elements of the data records returned by the second Web service.

For the data retrieved from weather reporting web services, the recall achieved by AReXS is significantly lower than the precision; that is, AReXS wasn’t wrong when it claimed a match, but there were several matches that it failed to recognise, namely
the fields labelled ‘Sky’ and ‘Visibility’. After running this experiment, a closer look at the data reveal that these fields were often empty. Because ARexs (and the original EBFM) depend on the contents of fields, empty or blank fields just can’t be overcome. In situations with a high rate of missing data, however, it is arguable as to the value of identifying matches; those matches would arguably be of little value as the fields’ contents themselves are inconsistent. Nonetheless, this is a weakness that structural or label-based ontology alignment algorithms don’t suffer.

6.3 Stock quotes

For a second experiment, the domain of share-market stock prices was chosen. Two web services were identified that provide information about the price of the stock of a given publicly-listed company: *Delayed Stock Quotes*, provided by CDYNE Corporation [18], and *Xignite Quotes*, provided by Xignite [140]. Examples of the records returned by these two web services can be found in Figures 6.5 and 6.6.

Twelve stocks were chosen and their records were retrieved from both stock price reporting web services. Examination of both the field names and the data led to the determination of correct matches, shown in Table 6.7. The data sets were processed

Figure 6.5: Example record collected from *Delayed Stock Quotes*
with the AReXS algorithm, producing the results presented in Table 6.8. Again, filtering for the strongest score per field is necessary and sensible, reducing the results to those shown in Table 6.9. A number of the scores are weak enough that some thresholding is required: choosing 0.7 includes all the correct matches and excludes the single remaining incorrect match (see Table 6.10. As will be seen later, this threshold value depends to some degree on the particular data, which is unsatisfying. Regardless, after filtering, the experiment results achieved are shown in Figure 6.11.

Certainly a degree of processing was required in order to achieve these results, but no more was required than simple selection of the strongest candidate for each field and a threshold to eliminate the single outlier.

### 6.4 Second-hand car classified advertisements

The final domain to which the AReXS algorithm was applied was that of second-hand car advertisements, as found on web sites. Classified advertisements are one of the more popular and successful uses that people have found for the Internet so far, probably due to the fact that they leverage the Internet’s strengths of convenient communication and information retrieval to bring together people with common interests while avoiding its weaknesses of poor security and privacy by conducting actual transactions off-line (although auction sites are also popular on the Internet, for high-value items such as cars, classified advertisement sites tend to act more like match-makers than brokers,
Table 6.7: Hand-chosen field matches for stock price reporting web services

<table>
<thead>
<tr>
<th>Field Match</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompanyName</td>
<td>Name</td>
</tr>
<tr>
<td>ChangePercent</td>
<td>PercentChange</td>
</tr>
<tr>
<td>DayHigh</td>
<td>High</td>
</tr>
<tr>
<td>DayLow</td>
<td>Low</td>
</tr>
<tr>
<td>FiftyTwoWeekRange</td>
<td>High52Weeks</td>
</tr>
<tr>
<td>LastTradeAmount</td>
<td>Last</td>
</tr>
<tr>
<td>OpenAmount</td>
<td>Open</td>
</tr>
<tr>
<td>PrevCls</td>
<td>PreviousClose</td>
</tr>
<tr>
<td>StockChange</td>
<td>Change</td>
</tr>
<tr>
<td>StockVolume</td>
<td>Volume</td>
</tr>
</tbody>
</table>

Table 6.8: AReXS field matches for stock price reporting web services

<table>
<thead>
<tr>
<th>Field Match</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompanyName</td>
<td>Name</td>
</tr>
<tr>
<td>ChangePercent</td>
<td>PercentChange</td>
</tr>
<tr>
<td>DayHigh</td>
<td>High</td>
</tr>
<tr>
<td>DayLow</td>
<td>Low</td>
</tr>
<tr>
<td>FiftyTwoWeekRange</td>
<td>High52Weeks</td>
</tr>
<tr>
<td>LastTradeAmount</td>
<td>Last</td>
</tr>
<tr>
<td>OpenAmount</td>
<td>Open</td>
</tr>
<tr>
<td>PrevCls</td>
<td>PreviousClose</td>
</tr>
<tr>
<td>StockChange</td>
<td>Change</td>
</tr>
<tr>
<td>StockVolume</td>
<td>Volume</td>
</tr>
</tbody>
</table>

106
### Table 6.9: AReXS field matches for stock price reporting web services filtered for best match per field

<table>
<thead>
<tr>
<th>Delayed Stock Quotes</th>
<th>Xignite Quotes</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompanyName</td>
<td>Name</td>
<td>0.8697</td>
</tr>
<tr>
<td>ChangePercent</td>
<td>PercentChange</td>
<td>0.7250</td>
</tr>
<tr>
<td>DayHigh</td>
<td>High</td>
<td>0.8944</td>
</tr>
<tr>
<td>DayLow</td>
<td>Low</td>
<td>0.9033</td>
</tr>
<tr>
<td>FiftyTwoWeekRange</td>
<td>High52Weeks</td>
<td>0.7669</td>
</tr>
<tr>
<td>LastTradeAmount</td>
<td>Last</td>
<td>0.9209</td>
</tr>
<tr>
<td>MktCap</td>
<td>High52Weeks</td>
<td>0.3175</td>
</tr>
<tr>
<td>OpenAmount</td>
<td>Open</td>
<td>0.9209</td>
</tr>
<tr>
<td>PrevCls</td>
<td>PreviousClose</td>
<td>0.9209</td>
</tr>
<tr>
<td>StockChange</td>
<td>Change</td>
<td>0.7849</td>
</tr>
<tr>
<td>StockVolume</td>
<td>Volume</td>
<td>0.8092</td>
</tr>
</tbody>
</table>

### Table 6.10: AReXS field matches for stock price reporting web services filtered for best match per field and threshold of 0.7

<table>
<thead>
<tr>
<th>Delayed Stock Quotes</th>
<th>Xignite Quotes</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompanyName</td>
<td>Name</td>
<td>0.8697</td>
</tr>
<tr>
<td>ChangePercent</td>
<td>PercentChange</td>
<td>0.7250</td>
</tr>
<tr>
<td>DayHigh</td>
<td>High</td>
<td>0.8944</td>
</tr>
<tr>
<td>DayLow</td>
<td>Low</td>
<td>0.9033</td>
</tr>
<tr>
<td>FiftyTwoWeekRange</td>
<td>High52Weeks</td>
<td>0.7669</td>
</tr>
<tr>
<td>LastTradeAmount</td>
<td>Last</td>
<td>0.9209</td>
</tr>
<tr>
<td>OpenAmount</td>
<td>Open</td>
<td>0.9209</td>
</tr>
<tr>
<td>PrevCls</td>
<td>PreviousClose</td>
<td>0.9209</td>
</tr>
<tr>
<td>StockChange</td>
<td>Change</td>
<td>0.7849</td>
</tr>
<tr>
<td>StockVolume</td>
<td>Volume</td>
<td>0.8092</td>
</tr>
</tbody>
</table>

**Precision:** 10/10 = 100%

**Recall:** 10/10 = 100%

### Table 6.11: Precision and recall of AReXS field matches for stock price reporting web services considering only strongest match per field with threshold of 0.7
<Autotrader>
  <car>
    <make_model>Toyota Corolla SECA</make_model>
    <yr>1993</yr>
    <kms>140,000</kms>
    <price>$9,995</price>
    <state>WA</state>
    <media>?</media>
    <dealer></dealer>
  </car>
</Autotrader>

**Figure 6.7:** Example record from *Autotrader*

bringing together interested sellers and buyers who then meet and make their own arrangements for sales and purchases). Five popular web sites were chosen:

- *Autotrader* - www.autotrader.com.au
- *Autoweb* - www.autoweb.com.au
- *Drive* - www.drive.com.au
- *Ebay* - www.ebay.com.au

**Collecting the data**

The search function of each site, an example of which is shown in Figure 6.8, was used to collect 100 current entries, creating a database of 500 car descriptions. The web sites typically presented the car descriptions in the form of tables, like that shown in Figure 6.9. This data was then parsed by custom written information agents to create pseudo-XML documents. The descriptions of cars were thereby transformed into documents of the form shown in Figure 6.7.

The data presented by the classified advertisement web sites was not always in the form of a simple table — Figures 6.10, 6.11 and 6.12 show different styles of presentation. Regardless, the use of task-oriented information agents to extract the required data from the web pages made collecting the data into a consistent form quite simple.
Where possible, the mark-up tag names used in the XML representations of the five sets of collected data were taken directly from the original information sources. Where no appropriate label was discernible for a field, a tag beginning with `Unknown` was applied; this has no impact on the performance of the AReXS algorithm as neither it nor the original EBFM algorithm use field or tag labels in any way. Regardless, the labels aid manual analysis of the data, although they can get in the way at times, as will be discussed later.

While several of the results pages presented by the web sites were quite tabular (such as those in Figures 6.9 and 6.10) and it was very easy to read a schema or ontology for the data directly from the column headings (see Figures 6.13 and 6.14\(^2\)), other web sites did not present such a straight forward format, notably those in Figures 6.11 and 6.12 — the representations chosen for these sources are shown in Figure 6.15 and 6.17. In these cases care was taken to maintain the apparent structure of the data as presented during the transformation to the XML representation.

\(^2\)For reasons related to printing, some of the figures in this article have had to be updated; some of the web sites involved had changed their design since the original analysis was performed and consequently look slightly different. Although none of the changes were significant, the reader may notice minor differences in field names and contents
**Used Vehicles**

- **State**: All States
- **Section**: Used Vehicles
- **Category**: Small, Medium, Family & Prestige
- **Make**: All Makes
- **Model**: All Models
- **Price Range**: $5000 to $10000

**Advanced Search Options**

- **Year Range**: __________ to __________
- **Source**: Include Private Classifieds, Include Commercial Classifieds, Include Dealer Online Ads
- **Age of Ads**: All Ads
- **Photos**: Photo Ads Only
- **Keyword/s**: __________
- **Details**: All Fuel Types __________, All Trans Types __________, All Body Types __________, All Drive Types __________

**Search**

* = mandatory field

**Figure 6.8**: Search interface for the Autotrader web site
### CHAPTER 6. APPLYING AREXS TO REAL, MESSY INFORMATION

<table>
<thead>
<tr>
<th>SAVE</th>
<th>PHOTO</th>
<th>MAKE &amp; MODEL</th>
<th>YR</th>
<th>KMS</th>
<th>REGION</th>
<th>STATE</th>
<th>PRICE</th>
<th>SELLER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NISSAN PULSAR manual</td>
<td>1993</td>
<td>107,000</td>
<td>Illawarra</td>
<td>NSW</td>
<td>$7,800</td>
<td>Private</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NISSAN SKYLINE manual coupe</td>
<td>1987</td>
<td>107,000</td>
<td>Gold Coast</td>
<td>QLD</td>
<td>$8,000</td>
<td>Private</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NISSAN BLUEBIRD manual</td>
<td>1995</td>
<td>107,000</td>
<td>Nth Shore</td>
<td>NSW</td>
<td>$7,850</td>
<td>Private</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOYOTA STARLET</td>
<td>1997</td>
<td>107,000</td>
<td>West Subs</td>
<td>NSW</td>
<td>$8,500</td>
<td>Private</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOLDEN ACCLAIM automatic sedan</td>
<td>1996</td>
<td>107,000</td>
<td>Hawkesbury</td>
<td>NSW</td>
<td>$8,000</td>
<td>Private</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VOLVO 850 automatic sedan</td>
<td>1992</td>
<td>107,000</td>
<td>Nth Shore</td>
<td>NSW</td>
<td>$9,950</td>
<td>Private</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FORD FALCON EL automatic station wagon</td>
<td>1996</td>
<td>107,000</td>
<td>Bris Metro</td>
<td>QLD</td>
<td>$9,500</td>
<td>Private</td>
</tr>
</tbody>
</table>

**Figure 6.9:** Car descriptions presented by the Autotrader web site
<table>
<thead>
<tr>
<th>Year</th>
<th>Make/Model</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>TOYOTA COROLLA</td>
<td>WHITE HATCHBACK AUTO</td>
<td>$8,990</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Location: CAPALABA, QLD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dealer Ad: Listed 07 Mar 2005</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>More details</td>
<td>Enquire now</td>
</tr>
<tr>
<td>2000</td>
<td>FORD FESTIVA GLXi</td>
<td>BURGUNDY HATCHBACK MANUAL</td>
<td>$8,990</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Odometer: 41,000 kms</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Location: OAKLEIGH, VIC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dealer Ad: Listed 07 Mar 2005</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>More details</td>
<td>Enquire now</td>
</tr>
<tr>
<td>1990</td>
<td>BMW 3 18i</td>
<td>BRONZE BEIGE SEDAN AUTO</td>
<td>$8,999</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Odometer: 203,048 kms</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Location: MOONEE PONDS, VIC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dealer Ad: Listed 07 Mar 2005</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>More details</td>
<td>Enquire now</td>
</tr>
<tr>
<td>1989</td>
<td>BMW 5 35i EXECUTIVE PACK</td>
<td>WHITE SEDAN AUTO</td>
<td>$9,999</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Odometer: 259,000 kms</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Location: MOONEE PONDS, VIC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dealer Ad: Listed 07 Mar 2005</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>More details</td>
<td>Enquire now</td>
</tr>
</tbody>
</table>

**Figure 6.10:** Car descriptions presented by the Autoweb web site
Figure 6.11: Car descriptions presented by the Carsales web site
**Figure 6.12:** Car descriptions presented by the Ebay website

<table>
<thead>
<tr>
<th>Item Title</th>
<th>PayPal</th>
<th>Price</th>
<th>Bids</th>
<th>Time Left</th>
</tr>
</thead>
<tbody>
<tr>
<td>PONTIAC TRANSAM 1980-</td>
<td>AU $12,000.00</td>
<td>-</td>
<td>14h 49m</td>
<td></td>
</tr>
<tr>
<td>STUNNER!!! GREAT PRICE!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOYOTA LANDCRUISER 78 SERIES 11 SEAT TROOPCARRIER</td>
<td>AU $34,900.00</td>
<td>-</td>
<td>17h 23m</td>
<td></td>
</tr>
<tr>
<td>SUPERB OFF ROAD OR GENERAL VEHICLE FOR CAMPING, TOWING,</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMW 525i, 1989 sedan</td>
<td>AU $10,000.00</td>
<td></td>
<td>21h 19m</td>
<td></td>
</tr>
<tr>
<td>1968 XTGT FALCON</td>
<td>AU $25,000.00</td>
<td>1</td>
<td>1d 00h 56m</td>
<td></td>
</tr>
<tr>
<td>Chevrolet Trail Blazer 2000 Luxury 4WD/SUV 54,000</td>
<td>AU $37,000.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLMS</td>
<td>AU $29,500.00</td>
<td>-</td>
<td>1d 02h 29m</td>
<td></td>
</tr>
<tr>
<td>Rare US Chevrolet Luxury SUV, Converted to Right Hand D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.5 On the correctness of semantic matches

The data having been collected and transformed into a processable format, the next step was to identify the correct matches between each pair of information sources. This proved much more difficult than for the previous experiments, due to the significantly greater complexity and variety of the data collected from the second-hand care sales web sites.

For the data collected from the domains of weather reports, stock price quotes and film screening sessions, identifying semantically equivalent data across information sources was quite straightforward. The matching required typically fell into one of two categories: either the field labels were missing or different but the field contents were very similar or identical, or the field contents were somewhat similar but the field labels were very similar or identical. The data collected from the second-hand car advertisement information sources, however, proved to be much more complex and confusing. Reading through the retrieved data quickly revealed several characteristics that distinguished the second-hand car advertisement data from the weather reports and stock price quotes.

The first such characteristic was that the data presented on the second-hand car advertisement web sites seemed to be much less well-formed and consistent than the previous data. Within individual fields, information was specified at varying levels of specificity and detail. For example, the ‘description’ field of the AutoWeb information source (see Figure 6.14 included entries with three different levels of specificity: ‘Mitsubishi Magna’, ‘Mitsubishi Magna TR’ and ‘Mitsubishi Lancer GLI CE’. Similarly, the ‘make_model’ field of the Autotrader information source (see Figure 6.13) contained entries consisting of ‘Hyundai Excel’ as well as of ‘Hyundai Excel JX’.

Some records included certain information in a field that other records omitted, such as in the ‘make_model’ field of the Autotrader information source, which contained entries such as ‘Holden Commodore VP’ but also ‘Holden Commodore VN SEDAN’.

Even less consistent were the contents of the ‘features’ and ‘description’ fields of the Drive information source (see Figure 6.16), the ‘Unknown3’ field of the CarSales information source and the ‘CAR’ field of the Ebay information source (see Figure 6.17), all of which had no apparent structure at all.

Finally, a number of fields frequently contained empty records:

- the ‘KMS’, ‘YR’ and ‘DEALER’ fields of the Autotrader information source were empty for 40%, 16% and 14% of the records collected,
• the ‘COLOUR’, ‘BODY_TYPE’ and ‘LOCATION’ fields of the AutoWeb information source were empty for 40%, 23% and 21% of the records collected,

• the ‘Unknown3’ field of the CarSales information source was empty for 20% of the records collected,

• the ‘colour’, ‘kms’, ‘body’, ‘features’, ‘trans’, ‘Unknown1’, ‘year’ and ‘description’ fields of the Drive information source were empty for 71%, 54%, 42%, 31%, 26%, 21%, 16% and 5% of the records collected, and

• the ‘PRICE’ field of the Ebay information source was empty for 4% of the records collected.

These four phenomena — varying degrees of specificity, occasional inclusion of certain information, apparent absence of structure and incomplete records — all suggest that much of the data present in these information sources was not created by experienced professional knowledge engineers, but by untrained people who, given the nature of classified advertisements, were ordinary members of the general public.

The second characteristic that distinguished this data from that presented previously was the number of fields included in each information source. It is unclear from the original EBFM description [56] whether each information source used contained fields representing all of the information present in other information sources of the same domain. For the information sources presented earlier in this chapter, a common feature is that each source tended to contain at least the fields present in the others, so that any missing fields were confined to only one source — in other words, one source was a super-set of the other (for instance, the Delayed Stock Quotes information source contains the fields ‘ChangePercent’, ‘MktCap’ and ‘PE’ which have no equivalent in the Xignite Quotes information source but the Xignite Quotes source doesn’t contain any fields that are not in the Delayed Stock Quotes source — compare Figures 6.5 and 6.6).

Since there can be no meaningful matches involving any of these “extra” fields, their presence (or absence in other sources) had no impact on the effectiveness of the AReXS algorithm as measured by the precision and recall metrics, which do not take into consideration data which is present in one source but not another. For the information sources found in the domain of second-hand car advertisements, a number of such fields were noticed, such as the fields labelled ‘trans’ and ‘ad_date’ in the Drive information source, the fields labelled ‘bids’ and ‘time_left’ in the Ebay source and the field labelled ‘location’ in the AutoWeb source, and these fields similarly had no impact on the effectiveness of the AReXS algorithm as measured by the precision and recall metrics.
CHAPTER 6. APPLYING AREXS TO REAL, MESSY INFORMATION

However, taken as a whole across the five sites included in the experiment, there were a number of fields that appear in several but not all of the information sources. For example, fields representing the colour of the cars described appear in the *AutoWeb* and *Drive* information sources but not in the other three sources. Fields representing the kilometres travelled by the cars described are present in the *Autotrader* and *Drive* information sources but not in the other three sources. This inconsistency across the information sources meant that the expected number of field matches was often very low, which, due to the nature of the precision and recall metrics, means that any mistakes are heavily punished — missing one out of three correct results gives a recall score of 66%, which looks little better than random guessing.

The third characteristic that distinguished the data in this experiment from that in the earlier experiments was that several fields represented information that in other sources was divided into two or more fields. For instance, consider the ‘Unknown1’ field from the *CarSales* information source, some instances of which are shown in Figure 6.15, and the ‘description’ field from the *Drive* information source, shown in Figure 6.16. Both of these fields contain data that looks like the contents of separate fields in other sources. The ‘Unknown1’ field of the *CarSales* source consistently contains what can only be interpreted as a year: compare ‘1986’, ‘1993’, ‘1997’, ‘1984’ and ‘1992’ from the records in Figure 6.15 to ‘1992’, ‘1996’, ‘1995’, ‘1994’, ‘1996’ and ‘1994’ taken from the ‘YEAR’ field of the *AutoWeb* source as shown in Figure 6.14 and ‘1993’, ‘1992’, ‘1991’ extracted from the ‘year’ field of the *Drive* source as in Figure 6.16.

The ‘Unknown1’ field of the *CarSales* source also consistently contains what looks like a manufacturer/model designation: compare ‘MAZDA RX7 FC1031 Limited’, ‘FORD LASER KH S’, ‘PROTON SATRIA GLi’, ‘MERCEDES 230E W123’ and ‘FORD FALCON EB II GLi’ (again from Figure 6.15) to ‘MITSUBISHI MAGNA’, ‘FORD FALCON’, ‘HOLDEN BARINA SWING’, ‘SUBARU SPORTSWAGON’ and ‘MITSUBISHI LANCER GLi CE’ from the ‘description’ field of the *AutoWeb* source in Figure 6.14 and ‘Mazda 323 Astina’, ‘Saab 9000’, ‘Subaru Liberty GX’, ‘Ford Falcon XH’ and ‘Ford Fairlane’ from the ‘MAKE_MODEL’ field of the *Autotrader* source.
<Autotrader>

<CAR>
  <MAKE_MODEL>Mazda 323 Astina</MAKE_MODEL>
  <YR>1992</YR>
  <KMS>140,000</KMS>
  <PRICE>$9,999</PRICE>
  <STATE>QLD</STATE>
  <DEALER>Ferrierr Qualit</DEALER>
</CAR>

<CAR>
  <MAKE_MODEL>Saab 9000</MAKE_MODEL>
  <YR>1987</YR>
  <KMS></KMS>
  <PRICE>$5,990</PRICE>
  <STATE>SA</STATE>
  <DEALER>Auto Credit Co</DEALER>
</CAR>

<CAR>
  <MAKE_MODEL>Subaru Liberty GX</MAKE_MODEL>
  <YR>1993</YR>
  <KMS></KMS>
  <PRICE>$9,990</PRICE>
  <STATE>VIC</STATE>
  <DEALER>Eastern Vehicle</DEALER>
</CAR>

<CAR>
  <MAKE_MODEL>Ford Falcon XH</MAKE_MODEL>
  <YR>1997</YR>
  <KMS></KMS>
  <PRICE>$9,999</PRICE>
  <STATE>VIC</STATE>
  <DEALER>Lucky Choice Ca</DEALER>
</CAR>

<CAR>
  <MAKE_MODEL>Ford Fairlane</MAKE_MODEL>
  <YR>1991</YR>
  <KMS>176,330</KMS>
  <PRICE>$9,000</PRICE>
  <STATE>NSW</STATE>
  <DEALER></DEALER>
</CAR>

</Autotrader>

Figure 6.13: Example records from the Autotrader source

118
<AutoWeb>
  <VEHICLE>
    <YEAR>1992</YEAR>
    <DESCRIPTION>MITSUBISHI MAGNA</DESCRIPTION>
    <PRICE>$9,990</PRICE>
    <BODY_TYPE>SEDAN</BODY_TYPE>
    <COLOUR>WHITE/GREY</COLOUR>
    <LOCATION>HAWTHORN, VIC</LOCATION>
  </VEHICLE>
  <VEHICLE>
    <YEAR>1996</YEAR>
    <DESCRIPTION>FORD FALCON</DESCRIPTION>
    <PRICE>$9,200</PRICE>
    <BODY_TYPE>WAGON</BODY_TYPE>
    <COLOUR></COLOUR>
    <LOCATION>NEWCASTLE AND CENTRAL</LOCATION>
  </VEHICLE>
  <VEHICLE>
    <YEAR>1995</YEAR>
    <DESCRIPTION>HOLDEN BARINA SWING</DESCRIPTION>
    <PRICE>$6,995</PRICE>
    <BODY_TYPE></BODY_TYPE>
    <COLOUR></COLOUR>
    <LOCATION>NEWCASTLE AND CENTRAL</LOCATION>
  </VEHICLE>
  <VEHICLE>
    <YEAR>1994</YEAR>
    <DESCRIPTION>SUBARU SPORTSWAGON</DESCRIPTION>
    <PRICE>$7,990</PRICE>
    <BODY_TYPE></BODY_TYPE>
    <COLOUR></COLOUR>
    <LOCATION>SYDNEY</LOCATION>
  </VEHICLE>
  <VEHICLE>
    <YEAR>1996</YEAR>
    <DESCRIPTION>MITSUBISHI LANCER GLI CE</DESCRIPTION>
    <PRICE>$9,990</PRICE>
    <BODY_TYPE>COUPE</BODY_TYPE>
    <COLOUR>METALLIC BLUE</COLOUR>
    <LOCATION></LOCATION>
  </VEHICLE>
</AutoWeb>

Figure 6.14: Example records from the AutoWeb source
Figure 6.15: Example records from the CarSales source
CHAPTER 6. APPLYING AREXS TO REAL, MESSY INFORMATION

$<$Drive$>
  $<$CAR$>
    $<$Unknown1$>$MERCEDES-BENZ 380$</Unknown1$>
    $<$Unknown2$>$9500$</Unknown2$>
    $<$year$>$0$</year$>
    $<$kms$>$ 0$</kms$>
    $<$trans$>$</trans$>
    $<$body$>$</body$>
    $<$colour$>$</colour$>
    $<$ad_date$>$03/05/2003$</ad_date$>
    $<$features$>$Anti lock braking, Alarm system, Cruise control, Radio cassette...$</features$>
    $<$description$>$MERCEDES 380SEL, Diamond Blue, blue lther, recond motor, Alpine stereo, alarm, ABS, sunrf, cruise, reg 1/04 YDQ756. $9500 ono 0439 311 767. $</description$>
    $<$location$>$The Sydney Morning Herald$</location$>
  $<$/CAR$>
$<$/Drive$>

$<$Drive$>
  $<$CAR$>
    $<$Unknown1$>$HOLDEN COMMODORE$</Unknown1$>
    $<$Unknown2$>$7250$</Unknown2$>
    $<$year$>$1992$</year$>
    $<$kms$>$175000$</kms$>
    $<$trans$>$</trans$>
    $<$body$>$</body$>
    $<$colour$>$</colour$>
    $<$ad_date$>$03/05/2003$</ad_date$>
    $<$features$>$</features$>
    $<$description$>$COMMODORE VP 1992 Series II, TSU-387, regd Nov ’03, new paint, tyres, very good cond, 175,000 ks, $7250 ono. 4965 9112. $</description$>
    $<$location$>$The Newcastle Herald and Post$</location$>
  $<$/CAR$>
$<$/Drive$>

Figure 6.16: Example records from the Drive source
Figure 6.17: Example records from the *Ebay* source
6.6 Surveying human judgements

Each of the five sets of car descriptions were paired and given to AReXS to reconcile, producing a collection of mappings between fields based on any evidence that AReXS could find to claim conceptual correspondences between the sets. For each pair of information sources, a set of correct matches had to be chosen.

The degree of complexity and inconsistency in the data sets for this experiment made choosing the ‘correct’ outcomes for the AReXS algorithm difficult. After making my own determinations, it was clear that many of the judgements were quite subjective. To overcome this problem it was decided to collect opinions from a number of other people and create an average assessment of which fields should be matched. A questionnaire was created, which can be found in Appendix 8.2. Six people completed the questionnaire, which is not strictly a significant number of respondents, but is certainly better than one person’s opinion alone. The responses have been summarised in Table 6.12.

The respondents were asked to identify matches between fields from each pair of information sources, and to indicate their confidence in the match by scoring it between 0 and 1, with 1 indicating that they consider the fields to be completely equivalent. Generally, the respondents tended to agree, but for a number of the more ambiguous fields there was a wide range of responses. As can be seen in Table 6.12, a number of potential matches between fields were identified by only a few of the respondents.
<table>
<thead>
<tr>
<th>Sources</th>
<th>Autotrader</th>
<th>AutoWeb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YR</td>
<td>YEAR</td>
<td>R1 0.8 R2 1 R3 1 R4 1 R5 1 R6 1</td>
</tr>
<tr>
<td>YEAR</td>
<td>YR</td>
<td>R1 0.8 R2 1 R3 1 R4 1 R5 1 R6 1</td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>DESCRIPTION</td>
<td>R1 1 R2 1 R3 0.5 R4 0.95 R5 0.8</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>MAKE_MODEL</td>
<td>R1 1 R2 1 R3 0.75 R4 1 R5 0.8</td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
<td>R1 1 R2 1 R3 1 R4 1 R5 1 R6 1</td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
<td>R1 1 R2 1 R3 1 R4 1 R5 1 R6 1</td>
</tr>
<tr>
<td>STATE</td>
<td>LOCATION</td>
<td>R1 0.5 R2 0.8 R3 0.2 R4 0.5 R5 0.8 R6 0.6</td>
</tr>
<tr>
<td>LOCATION</td>
<td>STATE</td>
<td>R1 0.0 R2 0.8 R3 0.2 R4 0.0 R5 0.0 R6 0.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Autotrader</td>
<td>CarSales</td>
<td></td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>Unknown1</td>
<td>R1 0.5 R2 0.5 R3 1 R4 0.1 R5 0.5 R6 0.2</td>
</tr>
<tr>
<td>Unknown1</td>
<td>MAKE_MODEL</td>
<td>R1 0.75 R2 0.5 R3 0.9 R4 0.1 R5 0.8 R6 0.2</td>
</tr>
<tr>
<td>YR</td>
<td>Unknown1</td>
<td>R1 0.5 R2 0.3 R3 0 R4 0.1 R5 0 R6 0</td>
</tr>
<tr>
<td>Unknown1</td>
<td>YR</td>
<td>R1 0.5 R2 0 R3 0.5 R4 0 R5 0 R6 0</td>
</tr>
<tr>
<td>KMS</td>
<td>Unknown1</td>
<td>R1 0.5 R2 0.5 R3 0 R4 0.1 R5 0.5 R6 0</td>
</tr>
<tr>
<td>Unknown1</td>
<td>KMS</td>
<td>R1 0.5 R2 0 R3 0 R4 0 R5 0.8 R6 0</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
<td>R1 1 R2 1 R3 0 R4 1 R5 1 R6 1</td>
</tr>
<tr>
<td>Unknown2</td>
<td>PRICE</td>
<td>R1 1 R2 0.8 R3 1 R4 1 R5 1 R6 1</td>
</tr>
<tr>
<td>STATE</td>
<td>Unknown4</td>
<td>R1 0.5 R2 0.8 R3 0.8 R4 0.5 R5 0.8 R6 0.6</td>
</tr>
<tr>
<td>Unknown4</td>
<td>STATE</td>
<td>R1 0.75 R2 0.8 R3 0.8 R4 0.5 R5 0.3 R6 0.6</td>
</tr>
</tbody>
</table>

*Table continued on next page...*
### CHAPTER 6. APPLYING AREXS TO REAL, MESSY INFORMATION

<table>
<thead>
<tr>
<th>Sources</th>
<th>Survey Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Autotrader</strong></td>
<td><strong>Drive</strong></td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>Unknown1</td>
</tr>
<tr>
<td>Unknown1</td>
<td>MAKE_MODEL</td>
</tr>
<tr>
<td>YR</td>
<td>year</td>
</tr>
<tr>
<td>year</td>
<td>YR</td>
</tr>
<tr>
<td>KMS</td>
<td>kms</td>
</tr>
<tr>
<td>kms</td>
<td>KMS</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
</tr>
<tr>
<td>Unknown2</td>
<td>PRICE</td>
</tr>
<tr>
<td>STATE</td>
<td>location</td>
</tr>
<tr>
<td>location</td>
<td>STATE</td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>description</td>
</tr>
<tr>
<td>description</td>
<td>MAKE_MODEL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Autotrader</th>
<th>Ebay</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKE_MODEL</td>
<td>CAR</td>
</tr>
<tr>
<td>CAR</td>
<td>MAKE_MODEL</td>
</tr>
<tr>
<td>YR</td>
<td>CAR</td>
</tr>
<tr>
<td>CAR</td>
<td>YR</td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
</tr>
</tbody>
</table>

*Table continued on next page...*
## Toward Semantic Interoperability for Software Systems

### Sources Survey Responses

<table>
<thead>
<tr>
<th>Source</th>
<th>CarSales</th>
<th>AutoWeb</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>Unknown1</td>
<td>0.5 0.5 0.2 0.1 0.5 0</td>
</tr>
<tr>
<td>Unknown1</td>
<td>YEAR</td>
<td>0.5 0.5 0.5 0.1 0.9 0</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Unknown1</td>
<td>0.5 0.5 1 0.1 0.5 0.2</td>
</tr>
<tr>
<td>Unknown1</td>
<td>DESCRIPTION</td>
<td>0.75 0.5 1 0.1 0.9 0.2</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
<td>1 1 1 1 1 1</td>
</tr>
<tr>
<td>Unknown2</td>
<td>PRICE</td>
<td>1 1 1 1 1 1</td>
</tr>
<tr>
<td>BODY_TYPE</td>
<td>Unknown1</td>
<td>0.5 0 0 0.1 0.5 0</td>
</tr>
<tr>
<td>Unknown1</td>
<td>BODY_TYPE</td>
<td>0.5 0 0 0 0.9 0</td>
</tr>
<tr>
<td>COLOUR</td>
<td>Unknown1</td>
<td>0.5 0 0 0.1 0.5 0</td>
</tr>
<tr>
<td>Unknown1</td>
<td>COLOUR</td>
<td>0.5 0 0 0 0 0</td>
</tr>
<tr>
<td>LOCATION</td>
<td>Unknown4</td>
<td>1 0.8 1 0.75 1 0.9</td>
</tr>
<tr>
<td>Unknown4</td>
<td>LOCATION</td>
<td>1 0.8 1 0.5 0.7 0.9</td>
</tr>
</tbody>
</table>

Table continued on next page...
### CHAPTER 6. APPLYING AREXS TO REAL, MESSY INFORMATION

<table>
<thead>
<tr>
<th>Sources</th>
<th>Survey Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoWeb</td>
<td></td>
</tr>
<tr>
<td>YEAR</td>
<td>R1</td>
</tr>
<tr>
<td>CAR</td>
<td>0.5</td>
</tr>
<tr>
<td>CAR</td>
<td>0.5</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>R1</td>
</tr>
<tr>
<td>CAR</td>
<td>0.75</td>
</tr>
<tr>
<td>CAR</td>
<td>0.75</td>
</tr>
<tr>
<td>PRICE</td>
<td>R1</td>
</tr>
<tr>
<td>PRICE</td>
<td>1</td>
</tr>
<tr>
<td>PRICE</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CarSales</th>
<th>Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown1</td>
<td>Unknown1</td>
</tr>
<tr>
<td>Unknown1</td>
<td>0.75</td>
</tr>
<tr>
<td>Unknown1</td>
<td>0.75</td>
</tr>
<tr>
<td>Unknown2</td>
<td>Unknown2</td>
</tr>
<tr>
<td>Unknown2</td>
<td>1</td>
</tr>
<tr>
<td>Unknown2</td>
<td>1</td>
</tr>
<tr>
<td>Unknown3</td>
<td>description</td>
</tr>
<tr>
<td>Unknown3</td>
<td>0</td>
</tr>
<tr>
<td>Unknown3</td>
<td>0.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CarSales</th>
<th>Ebay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown1</td>
<td>CAR</td>
</tr>
<tr>
<td>Unknown1</td>
<td>0.75</td>
</tr>
<tr>
<td>CAR</td>
<td>0.75</td>
</tr>
<tr>
<td>Unknown2</td>
<td>PRICE</td>
</tr>
<tr>
<td>Unknown2</td>
<td>1</td>
</tr>
<tr>
<td>PRICE</td>
<td>1</td>
</tr>
</tbody>
</table>

*Table continued on next page...*
It is notable that these results are not symmetrical; a number of the respondents made different decisions when presented with the information sources in different orders. The survey was designed to present each pair of information sources twice, in each possible order, so that the respondent would be asked, for instance, to match the data from *AutoWeb* with the data from *Drive*, and then later to match the data from *Drive* with the data from *AutoWeb*. As can be seen from the survey questionnaire in Appendix 8.2, other pairs of information sources were presented before any were repeated, so that the respondents would not immediately recall a pair the second time they encounter it. Although one respondent opted not to consider the repeated pairs, all of the other respondents performed the repeated comparison, with quite different results in several cases. For example, one respondent scored the ‘description’ and ‘CAR’ fields of the *Drive* and *Ebay* sources as 0.8 when presented in one order and 0 when reversed. Another respondent scored the ‘YR’ and ‘year’ fields of the *Autotrader* and *Drive* sources as 0.8 when presented in one order and 0 when reversed, and three respondents scored the ‘KMS’ and ‘kms’ fields of the same sources as 1 and 0 when the fields were reversed.

Ignoring the respondent who did not evaluate the repeated pairs, 38% of the identified match scores were scored differently when the same pair of information sources...
was presented in reverse. The average difference in score for the same pair of information sources was 0.18 (out of 1.0). This raises doubts as to the importance of an automated solution matching humans’ solutions — human decisions are invariably used as the ‘correct’ standard that machines should achieve, both in ontology alignment and other artificial intelligence areas, but this case is an example of the variance in human judgement.

To convert the collected survey responses into a consensus-agreed set of matches, two stages of filtering were applied. Firstly, the highest and lowest scores were discarded for each pair of fields; this accommodated several instances in which one of the respondents appeared to have overlooked a match that all the other respondents identified with the highest confidence. An example of this is the pair of fields ‘PRICE’ and ‘Unknown2’ in the sources Autotrader and CarSales, which five out of the six respondents identified with confidence of 1.0 but one respondent omitted completely. The remaining scores were then averaged.

Secondly, the scores for each pair of fields were averaged, so that the score given to a pair of fields when the data sets were presented in one order was combined with the score given when the order was reversed. This was done based on the assumption that any technique for semantic interoperability should be symmetrical, even though the human judgements varied depending on which data set was presented first.

Finally, the resulting score for each pair of fields was filtered by a threshold of 0.8; this allowed matches to be considered correct where one or two of the respondents were reluctant to identify with complete confidence, yet still clearly considered the match to be strong. An example of this is the pair of fields ‘MAKE\_MODEL’ and ‘DESCRIPTION’ in the sources Autotrader and AutoWeb, which three respondents identified with confidence 1.0 but the other respondents scored between 0.5 and 0.95. Although the choice of 0.8 as a threshold was not based on any established practice or convention, the highest scored pair of fields that fell below the 0.8 threshold was scored at 0.75, and the next below it was scored at 0.67, so it was felt that a legitimate argument could be made that there was a natural gap between scores above 0.8 and those below.

Nineteen pairs of fields passed the threshold, spread fairly evenly across the different pairs of information sources. The filtered set of scores is shown in Table 6.13.
# Toward Semantic Interoperability for Software Systems

Sources | Filtered survey responses
---|---
**Autotrader** | **AutoWeb**
MAKE\_MODEL | DESCRIPTION | 0.94
YR | YEAR | 1.00
PRICE | PRICE | 1.00

**Autotrader** | **CarSales**
PRICE | Unknown2 | 1.00

**Autotrader** | **Drive**
MAKE\_MODEL | Unknown1 | 1.00
PRICE | Unknown2 | 1.00

**Autotrader** | **Ebay**
MAKE\_MODEL | CAR | 0.88
PRICE | PRICE | 1.00

**AutoWeb** | **CarSales**
PRICE | Unknown2 | 1.00
LOCATION | Unknown4 | 0.89

**AutoWeb** | **Drive**
YEAR | year | 0.84
DESCRIPTION | Unknown1 | 1.00
PRICE | Unknown2 | 1.00

**AutoWeb** | **Ebay**
DESCRIPTION | CAR | 0.80
PRICE | PRICE | 1.00

**CarSales** | **Drive**
Unknown2 | Unknown2 | 1.00

**CarSales** | **Ebay**
Unknown2 | CAR | 1.00

*Table continued on next page...*
CHAPTER 6. APPLYING AREXS TO REAL, MESSY INFORMATION

Sources Filtered survey responses

<table>
<thead>
<tr>
<th>Drive</th>
<th>Ebay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown1</td>
<td>CAR</td>
</tr>
<tr>
<td>Unknown2</td>
<td>PRICE</td>
</tr>
</tbody>
</table>

**Table 6.13:** Correct field matches as chosen by the survey respondents after filtering

My own choices for fields that should be matched are presented in Table 6.14. There are two differences between my decisions and those of the survey respondents.

The first difference is based on the handling of incomplete data. I considered the ‘year’ and ‘kms’ fields of the Drive source to match the ‘YR’ and ‘KMS’ fields respectively of the Autotrader source whereas the survey respondents were not confident enough about these pairs of fields to declare them as matching. This is quite understandable since, as mentioned earlier, the ‘kms’ field of the Drive information source was empty for 54% of the records. This was reflected in the data presented for the survey, which consists for the Drive source of two out of four records containing an empty ‘kms’ field.

Less expectedly, the survey respondents didn’t consider the ‘year’ field of the Drive source to match the ‘YR’ field of the Autotrader source. All four of the Autotrader records and three of the four Drive records presented contained what seems obviously to be a year in their ‘year’ and ‘YR’ fields, and the survey respondents thought so when the records from the Autotrader source were presented first, but when the Drive records were presented first, the survey responses were drastically lower, with three out of six respondents not declaring a match with any confidence at all. This was possibly caused by the apparently invalid ‘0’ value in the first Drive record; the effect was to rule out this pair of fields as a desired match.

The second difference between my choices and the summarised survey responses relates to precision and specificity. Three pairs of fields were declared to match by the survey respondents that I had rejected:

- the ‘MAKE_MODEL’ and ‘CAR’ fields of the Autotrader and Ebay sources,
- the ‘DESCRIPTION’ and ‘CAR’ fields of the AutoWeb and Ebay sources, and

131
• the ‘Unknown1’ and ‘CAR’ fields of the Drive and Ebay sources.

The common element to these pairs of fields is the ‘CAR’ field of the Ebay information source, which appears to be an unstructured description of a car written by the car’s owner. I didn’t consider its contents to be either consistent or specific enough to match any of the ‘description’ fields from the other sources, as it often contains extraneous information at the same time that it omits important identifying information. For example, consider the records shown in Figure 6.17. Although one record’s ‘CAR’ field contains ‘Ford Fairlane 1991 NC’, which is a neat and clear identification of the car’s make and model and would match well with fields from several of the other information sources, other records contain ‘MITSUBISHI MAGNA TJ SPORTS 2001 MODEL’, ‘1985 Holden VK Commodore Executive Unreg’, which include information that doesn’t belong in a strict make and model identification, and even ‘MUST BE SEEN. SERIES II VR SS V8 5 SPEED’, which not only contains unclassifiable extra information but doesn’t actually specify any make or model, merely a designation within a particular model. Although most car enthusiasts would recognise that a ‘Series II VR SS’ is a Holden Commodore, I didn’t consider this field to be a good match for any of the other ‘description’ fields. The survey respondents, however, did consider it to be an acceptable match, albeit it weakly, as can be seen from the averaged scores presented in Table 6.13.

In the final analysis, I have decided to report results for this experiment using both my decisions as to the correct matches and those of the survey respondents. Having presented an analysis of the differences, and explaining the reasoning behind my own decisions, I hope that testing AReXS against both sets of “correct” results will provide an interesting contrast.

6.7 Results

The results generated by AReXS were significantly poorer for the messy data collected from second-hand car sales web sites than for the cleaner data obtained for the earlier experiments. Initially, the conditions applied to the earlier experiments were replicated for this data set, namely that the output from AReXS was filtered by a threshold of 0.7. The results of the experiment are shown in Table 6.15. The corresponding precision and recall scores measured against the averaged and filtered choices of the survey respondents can be seen in Figure 6.16, while Figure 6.17 shows the precision and recall measured against my choices.
### CHAPTER 6. APPLYING AREXS TO REAL, MESSY INFORMATION

<table>
<thead>
<tr>
<th>Sources</th>
<th>My choices after filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autotrader</td>
<td>AutoWeb</td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>YR</td>
<td>YEAR</td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
</tr>
<tr>
<td>Autotrader</td>
<td>CarSales</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
</tr>
<tr>
<td>Autotrader</td>
<td>Drive</td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>Unknown1</td>
</tr>
<tr>
<td>YR</td>
<td>year</td>
</tr>
<tr>
<td>KMS</td>
<td>kms</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
</tr>
<tr>
<td>Autotrader</td>
<td>Ebay</td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
</tr>
<tr>
<td>AutoWeb</td>
<td>CarSales</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
</tr>
<tr>
<td>LOCATION</td>
<td>Unknown4</td>
</tr>
<tr>
<td>AutoWeb</td>
<td>Drive</td>
</tr>
<tr>
<td>YEAR</td>
<td>year</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Unknown1</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
</tr>
<tr>
<td>BODY_TYPE</td>
<td>body</td>
</tr>
<tr>
<td>COLOUR</td>
<td>colour</td>
</tr>
<tr>
<td>AutoWeb</td>
<td>Ebay</td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
</tr>
<tr>
<td>CarSales</td>
<td>Drive</td>
</tr>
<tr>
<td>Unknown2</td>
<td>Unknown2</td>
</tr>
</tbody>
</table>

Table continued on next page...
### Table 6.14: My choices for correct field matches between second-hand car web sites after filtering

<table>
<thead>
<tr>
<th>Sources</th>
<th>My choices after filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>CarSales</td>
<td>Ebay</td>
</tr>
<tr>
<td>Unknown2</td>
<td>PRICE 1.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sources</th>
<th>Ebay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive</td>
<td></td>
</tr>
<tr>
<td>Unknown2</td>
<td>PRICE 1.00</td>
</tr>
</tbody>
</table>

**Sources**

<table>
<thead>
<tr>
<th>Sources</th>
<th>Autotrader</th>
<th>AutoWeb</th>
<th>ARExS Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKE_MODEL</td>
<td>DESCRIPTION</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>BODY_TYPE</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>LOCATION</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>YR</td>
<td>YEAR</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>KMS</td>
<td>PRICE</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>STATE</td>
<td>DESCRIPTION</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>STATE</td>
<td>LOCATION</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>DEALER</td>
<td>BODY_TYPE</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>DEALER</td>
<td>COLOUR</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>DEALER</td>
<td>LOCATION</td>
<td>0.06</td>
<td></td>
</tr>
</tbody>
</table>

**Sources**

<table>
<thead>
<tr>
<th>Sources</th>
<th>CarSales</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKE_MODEL</td>
<td>Unknown1</td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>Unknown4</td>
</tr>
<tr>
<td>YR</td>
<td>Unknown1</td>
</tr>
<tr>
<td>YR</td>
<td>Unknown3</td>
</tr>
<tr>
<td>KMS</td>
<td>Unknown2</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
</tr>
<tr>
<td>STATE</td>
<td>Unknown3</td>
</tr>
<tr>
<td>STATE</td>
<td>Unknown4</td>
</tr>
</tbody>
</table>

*Table continued on next page...*
## CHAPTER 6. APPLYING AREXS TO REAL, MESSY INFORMATION

<table>
<thead>
<tr>
<th>Sources</th>
<th>AReXS Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEALER</td>
<td>Unknown3</td>
</tr>
<tr>
<td>DEALER</td>
<td>Unknown4</td>
</tr>
</tbody>
</table>

### Autotrader

<table>
<thead>
<tr>
<th>Source</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKE_MODEL</td>
<td>Unknown1</td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>Unknown2</td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>body</td>
</tr>
<tr>
<td>MAKE_MODEL</td>
<td>description</td>
</tr>
<tr>
<td>YR</td>
<td>Unknown2</td>
</tr>
<tr>
<td>YR</td>
<td>year</td>
</tr>
<tr>
<td>YR</td>
<td>kms</td>
</tr>
<tr>
<td>YR</td>
<td>ad_date</td>
</tr>
<tr>
<td>YR</td>
<td>description</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
</tr>
<tr>
<td>PRICE</td>
<td>description</td>
</tr>
<tr>
<td>STATE</td>
<td>Unknown1</td>
</tr>
<tr>
<td>STATE</td>
<td>description</td>
</tr>
<tr>
<td>STATE</td>
<td>location</td>
</tr>
<tr>
<td>DEALER</td>
<td>Unknown1</td>
</tr>
<tr>
<td>DEALER</td>
<td>body</td>
</tr>
<tr>
<td>DEALER</td>
<td>colour</td>
</tr>
<tr>
<td>DEALER</td>
<td>description</td>
</tr>
</tbody>
</table>

### Autotrade

<table>
<thead>
<tr>
<th>Source</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKE_MODEL</td>
<td>CAR</td>
</tr>
<tr>
<td>YR</td>
<td>CAR</td>
</tr>
<tr>
<td>KMS</td>
<td>PRICE</td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
</tr>
</tbody>
</table>

### AutoWeb

<table>
<thead>
<tr>
<th>Source</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>Unknown1</td>
</tr>
<tr>
<td>YEAR</td>
<td>Unknown3</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Unknown1</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
</tr>
</tbody>
</table>

*Table continued on next page...*
<table>
<thead>
<tr>
<th>Sources</th>
<th>AReXS Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>BODY_TYPE</td>
<td>Unknown1</td>
</tr>
<tr>
<td>COLOUR</td>
<td>Unknown1</td>
</tr>
<tr>
<td>COLOUR</td>
<td>Unknown3</td>
</tr>
<tr>
<td>LOCATION</td>
<td>Unknown4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AutoWeb</th>
<th>Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>Unknown2</td>
</tr>
<tr>
<td>YEAR</td>
<td>year</td>
</tr>
<tr>
<td>YEAR</td>
<td>kms</td>
</tr>
<tr>
<td>YEAR</td>
<td>description</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Unknown1</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Unknown2</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>body</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>description</td>
</tr>
<tr>
<td>PRICE</td>
<td>Unknown2</td>
</tr>
<tr>
<td>PRICE</td>
<td>kms</td>
</tr>
<tr>
<td>PRICE</td>
<td>description</td>
</tr>
<tr>
<td>BODY_TYPE</td>
<td>Unknown1</td>
</tr>
<tr>
<td>BODY_TYPE</td>
<td>body</td>
</tr>
<tr>
<td>BODY_TYPE</td>
<td>description</td>
</tr>
<tr>
<td>COLOUR</td>
<td>colour</td>
</tr>
<tr>
<td>COLOUR</td>
<td>features</td>
</tr>
<tr>
<td>COLOUR</td>
<td>description</td>
</tr>
<tr>
<td>LOCATION</td>
<td>description</td>
</tr>
<tr>
<td>LOCATION</td>
<td>location</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AutoWeb</th>
<th>Ebay</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>CAR</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>CAR</td>
</tr>
<tr>
<td>PRICE</td>
<td>PRICE</td>
</tr>
<tr>
<td>BODY_TYPE</td>
<td>CAR</td>
</tr>
<tr>
<td>COLOUR</td>
<td>CAR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CarSales</th>
<th>Drive</th>
</tr>
</thead>
</table>

Table continued on next page...
### Table 6.15: AReXS filtered field matches for second-hand car sales web sites

<table>
<thead>
<tr>
<th>Sources</th>
<th>AReXS Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown1</td>
<td>Unknown1 0.83</td>
</tr>
<tr>
<td>Unknown1</td>
<td>year 0.78</td>
</tr>
<tr>
<td>Unknown1</td>
<td>kms 0.04</td>
</tr>
<tr>
<td>Unknown1</td>
<td>body 0.98</td>
</tr>
<tr>
<td>Unknown1</td>
<td>colour 0.70</td>
</tr>
<tr>
<td>Unknown2</td>
<td>Unknown2 0.73</td>
</tr>
<tr>
<td>Unknown2</td>
<td>kms 0.02</td>
</tr>
<tr>
<td>Unknown2</td>
<td>description 0.37</td>
</tr>
<tr>
<td>Unknown3</td>
<td>Unknown2 0.10</td>
</tr>
<tr>
<td>Unknown3</td>
<td>year 0.04</td>
</tr>
<tr>
<td>Unknown3</td>
<td>colour 0.06</td>
</tr>
<tr>
<td>Unknown3</td>
<td>features 0.13</td>
</tr>
<tr>
<td>Unknown3</td>
<td>description 0.16</td>
</tr>
<tr>
<td>Unknown4</td>
<td>colour 0.02</td>
</tr>
<tr>
<td>Unknown4</td>
<td>location 0.21</td>
</tr>
</tbody>
</table>

**CarSales**

<table>
<thead>
<tr>
<th>Ebay</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown1</td>
<td>CAR 0.16</td>
</tr>
<tr>
<td>Unknown2</td>
<td>PRICE 0.60</td>
</tr>
<tr>
<td>Unknown3</td>
<td>CAR 0.11</td>
</tr>
</tbody>
</table>

**Drive**

<table>
<thead>
<tr>
<th>Ebay</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown1</td>
<td>CAR 0.98</td>
</tr>
<tr>
<td>Unknown2</td>
<td>CAR 0.05</td>
</tr>
<tr>
<td>Unknown2</td>
<td>PRICE 0.21</td>
</tr>
<tr>
<td>year</td>
<td>CAR 0.53</td>
</tr>
<tr>
<td>body</td>
<td>CAR 0.70</td>
</tr>
<tr>
<td>colour</td>
<td>CAR 0.20</td>
</tr>
<tr>
<td>description</td>
<td>CAR 0.03</td>
</tr>
</tbody>
</table>
### Table 6.16:

<table>
<thead>
<tr>
<th>Source</th>
<th>Source</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autotrader</td>
<td>AutoWeb</td>
<td>75%</td>
<td>100%</td>
</tr>
<tr>
<td>Autotrader</td>
<td>CarSales</td>
<td>33%</td>
<td>100%</td>
</tr>
<tr>
<td>Autotrader</td>
<td>Drive</td>
<td>33%</td>
<td>50%</td>
</tr>
<tr>
<td>Autotrader</td>
<td>Ebay</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>AutoWeb</td>
<td>CarSales</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>AutoWeb</td>
<td>Drive</td>
<td>50%</td>
<td>100%</td>
</tr>
<tr>
<td>AutoWeb</td>
<td>Ebay</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>CarSales</td>
<td>Drive</td>
<td>50%</td>
<td>100%</td>
</tr>
<tr>
<td>CarSales</td>
<td>Ebay</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Drive</td>
<td>Ebay</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td><strong>Average of all sources</strong></td>
<td></td>
<td><strong>59%</strong></td>
<td><strong>65%</strong></td>
</tr>
</tbody>
</table>

#### 6.8 Discussion

I feel that these results are mixed. At face value, these are quite low precision and recall scores, especially when compared to other published results for automatic ontological reconciliation. On the other hand, the data collected for this experiment was deliberately sourced from messy records created in many cases by casual users, with no cleaning or filtering. Additionally, the extended EBFM algorithm was not tuned in any way for this data. I would question the choices of correct matches made by the survey participants, if it were not for the fact that the AReXS system performed better by their standards than when tested against my own choices.

As I noted earlier, there was considerable disagreement among the survey participants as to the correct results for many of the cases presented to them, not to mention the fact that many of the people surveyed nominated different “correct” results when presented with the data in reverse order. This strongly suggests that it is reasonable to claim that for some of the data used in this experiment, there really is no correct result, which certainly complicates the findings.

A further weakness of the survey is that only a small subset of the data used in the experiment was presented to the participants, and it possible that this data was in some way unrepresentative, although it was selected randomly from the experimental data set.
On the whole, I think that it is reasonable to draw two conclusions from this experiment: that the AReXS system, and indeed the original EBFM algorithm on which its algorithm is based, are a strong demonstration of the usefulness of instance-based approaches to reconciling semantically incompatible data sources, and that such an approach is not limited to the well-formed data to which it was originally applied, but is capable of handling significantly messier data. With that said, though, I have also demonstrated that there is a limit to the quality of data to which such an approach can be applied, and the results presented in this chapter show which types of real-world data can be tackled in this way, and which cannot.

Along with the results of adapting the EBFM algorithm to real, messy data, I want to make some observations regarding the matching of strings. In the previous chapter, I described a variety of techniques for calculating the similarity of pairs of strings. For the AReXS system described in this chapter, the $n$-gram technique was used, but a number of cases were revealed in which it is less than satisfactory. I would like to discuss these cases and consider how they might be better accommodated.

**Acronyms**

It is clear to a person with only limited domain knowledge that ‘IBM’ is likely an acronym for ‘International Business Machines’, and that ‘UK’ is likely an acronym for ‘United Kingdom’. In these cases, the domain from which some knowledge is required
is that of western cultural general knowledge, which can be obtained merely by living in the western world for a moderate length of time. An n-gram approach to string comparison, on the other hand, will find no correspondence between these strings. Whether this is positive or negative depends on the task for which the comparison is performed; for instance, for the information sources presented in this chapter, it is not clear that there are many acronyms that will be encountered and should be considered equivalent. Since cars are often referred to by their model name (e.g. “I just bought a Holden Commodore.”) and also by just the name of the company that produced them (e.g. “This is my new Holden.”), it is possible that some cars could be referred to by the acronym of their manufacturer. However, there seem to be no examples of car manufacturers that are commonly referred to by both their full name and an acronym; Bavarian Motorwerks (BMW) and General Motors Corporation (GMC) seem to be the only companies referred to by their acronym, and when referring to cars produced by these companies the full name is rarely used.

Of course, as noted, this discussion is closely tied to the example information sources presented in this chapter. By contrast, for example, reconciling geographical information seems likely to require that acronyms be treated as equivalent to full names, such as the locations referenced in some of the information sources presented, e.g. ‘NSW’ and ‘WA’, which should be treated as equivalent to the full names of the states for which they are acronyms, namely ‘New South Wales’ and ‘Western Australia’.

Abbreviations

The information sources presented in this chapter employ a number of abbreviations, most notably for locations (in a similar fashion to that described above for acronyms), such as ‘Vic’ for ‘Victoria’ and ‘Mel’ for ‘Melbourne’. A string comparison method that catered for these alternate representations of concepts would likely perform better than the implemented n-gram algorithm, but it is not clear whether such abbreviations should rightly be considered to be domain knowledge or could be catered for via general linguistic techniques. Interestingly, the most obvious abbreviations occur in the field names of the information sources, such as ‘kms’ for ‘kilometres’ and ‘yr’ for ‘year’. However, since the EBFM algorithm used in AReXS doesn’t consider the names of fields, considering or ignoring these abbreviations doesn’t alter the result of the reconciliation.
Sets

Because the EBFM algorithm on which AReXS is based focuses on individual instances that match strongly, fields whose values form a set of a small number of syntactically unrelated strings, such as colours, aren’t handled well. Although only one information source among those presented in this chapter included data about colours, it is easy to imagine this sort of information being more prominent in other domains. This is important because identifying matching fields by treating them as sets of elements could be significantly more effective than looking for pairs of matching instances from with the fields. For example, the sets \{‘red’, ‘blue’, ‘red’, ‘yellow’, ‘yellow’\} and \{‘blue’, ‘blue’, ‘yellow’, ‘green’, ‘red’\} might be more readily matched by set operations.

Units

A particular vulnerability of syntactic string comparison methods is when extra information is encoded in a string in such a way that requires domain knowledge to decode. Currencies are a good example — the string ‘$AU 599.95’ is almost certainly a lot more similar to the string ‘$599.95’ than any of the comparison methods described earlier in this chapter would realise.

Numeric values

Finally, the most obvious case in which string comparison falls down is when the strings being compared do not in fact represent string data at all, but are simply encoded numeric values. The values ‘1999’ and ‘2000’ are almost identical when they represent calendar years, yet syntactically have nothing in common apart from their length. Likewise, the values ‘-1’ and ‘2’, ‘99’ and 101’ and so on. Recognising and interpreting numbers for what they are, rather than treating all data as strings, would seem to be essential for any similarity function that will be expected to function appropriately in the face of real-world data.

In summary, in order to function with real data from heterogeneous sources, it is clear from the results presented here that a similarity function requires the ability to recognise and deal with a range of special cases. How best to do this in practice is not clear. There seems to be little recent research on this topic, although the work presented by Dhamankar et alia touches on the use of heuristics to identify data types and then apply specialised comparison functions [28], and work by Naumann et alia attempts to automatically determine a type, or ‘signature’, for elements or attributes in databases and information sources by applying feature analysis to the instances.
of the data source [93]. As a quick experiment along the same lines, I added two specialised field comparison functions to the AReXS system for handling numerical fields and allowing whole-word matching rather than $n$-gram matching. The results were, however, mixed: while some extra correct field matches were identified over and above those reported formally earlier in this chapter, which would improve the measured recall, a similar number of incorrect matches were also reported as the algorithms were confused by the particularly messy and inconsistent data in some of the data sets, meaning that the precision would be reduced. Overall, I feel that this merely strengthens the argument for small, purposive, heuristics to extend the usefulness of more general algorithms.

The use of such heuristics, sometimes called “data detectors” [5] to recognise and process particular types of information is a similar approach that is beginning to appear in more mainstream software applications, such as many e-mail applications and web sites which recognise e-mail addresses and URLs that have been typed as plain text in e-mail messages and mark them up as selectable links. Apple, Inc., provides a more sophisticated version in their Mail application, which attempts to recognise dates and times as typed in e-mail messages to allow them to be immediately added as events to the user’s calendar [5]. The Skype Internet telephony application from Skype Limited includes a plug-in for web browsers that recognises telephone numbers in web pages viewed and provides a contextual button to enable the user to immediately call that number through the Skype application [116]. The Google Maps service provided by Google, Inc., recognises and decodes street addresses as typed by users and translates them into longitude and latitude co-ordinates in order to present localised maps to users [45]. These small units of intelligence built-in to larger software systems seem to be converging with micro-formats, which I will discuss in the next chapter.

It seems likely to me that lightweight heuristic solutions such as these will be increasingly valuable in augmenting more general algorithms and techniques such as the extended EBFM algorithm presented here. Such approaches will be necessary at least until the more formal type tagging approach proposed by the Semantic Web project becomes widely accepted, and in the event that it does not, will remain necessary.
Chapter 7

Data Design Collaboration

The preceding chapters in this thesis have explored different approaches to addressing the problem of reconciling data whose description and representation has diverged as independent systems have been designed and implemented. The term ‘ontological reconciliation’ has been used to describe the process of bringing heterogeneously represented data back into a homogeneous form in order that they may be processed or manipulated seamlessly (as opposed to the term ‘ontology alignment’ which generally refers to the more narrowly focused exercise of constructing a mapping between two formal ontologies). It has been posited that the primary cause of the problem of semantic interoperability is that the process of designing a (machine-readable) representation for a given concept or datum is one of choice, as for any given concept there are many possible reasonable representations, and thus when individuals perform this design activity they will inevitably make different choices, for pragmatic, philosophical or experiential reasons. For example, a particular database management system might provide a specific facility for storing the currency of monetary amounts whereas other systems might require an interpretation of the integer or floating point field contents; an expert in the domain of automobiles might consider the number of gears in an automatic transmission to be significant enough to warrant inclusion in a description of a car, whereas a designer with less interest in the subject matter might not.

It has been argued throughout this thesis that these variations in choice of representation can be significantly difficult to reconcile once systems have been implemented and deployed. It is commonly accepted in the software engineering field that the cost of repairing faults in a system increases by orders of magnitude as a project progresses from requirements elicitation, through design and implementation, to deployment and maintenance, and yet it is commonly assumed that most interoperability will be at-
tempted after deployment, i.e. as a maintenance activity. As it is predominantly used today, the notion of ontology alignment presupposes that two or more ontologies have been fully designed, and software engineering experience tells us that this is the most expensive time to change the behaviour of a system\(^1\). Following this principle of dramatically increasing cost of modification, it is natural to imagine that the problem of semantic interoperability could be reduced if the divergence of representation of concepts and data could be reduced during the design of systems, rather than after their construction.

I would expect the most likely objection to this suggestion to be that interoperability is an opportunistic phenomenon, by which the value of two existing systems is increased by combining their functionality in a new way, a way that wasn’t foreseen when the individual systems were conceived. There are two rebuttals to this suggestion, the first being that after more than a decade of widespread public, commercial and governmental use of the Internet to connect more people, systems and information than have ever been connected before, it is myopic not to imagine that any new system might benefit from interoperating with other systems, and the second being that although the problem of interoperation exists precisely because individual systems are generally built without knowledge of the other systems with which they might interact in the future, this needn’t be the case.

### 7.1 Barriers to aligning data definitions during development

In order to reduce the divergence of representation of data during the design of systems, there would need to be some communication between the designers of different systems, and some sharing of proposed representations. There are several barriers to this sharing:

- systems are often designed at different times and schedules, and a project generally cannot delay specifying its data representations until another project has reached the same stage

- systems are often designed and developed at a distance from each other and therefore may not have the means to communicate effectively or even be aware of each other’s existence

\(^1\)It could be argued that creating mediating layers that translate between heterogeneous systems is not the same as modifying the systems themselves, and thus should not be as expensive, but experience suggests that inserting intermediary systems complicates interoperation and increases the points of possible failure, making such a proposal less than satisfactory as a long-term solution.
designers of systems may well not wish to disclose their data representations for commercial or other reasons

there may be unavoidable constraints on a project that prevent altering some or all aspects of a data representation

While the third and fourth obstacles are somewhat insurmountable in the context of design, being the result of choices made outside the purview of the system developers, a solution is proposed to the first two barriers a public repository of data representations: a public repository into which system designers can place the data representations used in their system and then browse and search other definitions, employing alignment and similarity techniques to identify semantically related classes and data structures from other projects.

The proposed system will follow the model of collaborative information repositories such as the dmoz open directory project, Wikipedia, del.icio.us, digg, The Code Project and SourceForge. Something that all of these projects have in common is the principle of consensus and collaboration leading to greater outcomes than could be achieved by individuals in isolation, and I think it is valuable to explore this characteristic before launching into a description of the proposed repository.

The dmoz open directory project aims to construct a catalogue of on-line resources, organised by categories to assist browsing and searching. Resources are discovered by individuals who then place them in whichever category they believe is most appropriate. The underlying philosophy of the project is that previous catalogues of on-line resources, such as the Yahoo! Directory, failed because there are too many resources to be identified and located by any one organisation. By contrast, spreading the workload across as many people as are motivated to contribute will arguably lead to a more complete work.

Wikipedia is an attempt to produce an encyclopaedia of knowledge and facts as agreed by thousands of equally-important contributors. Any entry for a topic can be modified and extended at any time by any person. The philosophy behind the project is similar to that of the dmoz open directory, in that the effort required to produce a complete and accurate encyclopaedia is too great for any single organisation, but also includes a strong element of democracy, in that each element of the work can be modified by any person and there is no authority that approves or rejects submissions. It has been argued that this is not in fact democratic at all, and I tend to agree: the goal of removing central authority has less to do with democracy than it does with a combination of anarchy and anti-patriarchal ideas. The only way that Wikipedia
provides for on-line ‘citizens’ to support a particular entry is by silently agreeing not to changed or remove it, and a single person can over-ride any number of opposers by changing an entry at any time. Regardless of socio-political arguments, Wikipedia is an impressive example of the potential of mass collaboration.

Del.icio.us and Digg capture and present to readers popular pieces of news, information and on-line resources. Del.icio.us aggregates its users’ web-browsing ‘bookmarks’, the list of web sites that each user has compiled and stored for future reference, and uses a ‘recommendation’ model to present to users new resources that the site’s algorithms calculate might be of interest. In this way the model is similar to that employed by on-line retailers such as Amazon and the social Internet radio station last.fm, which reason that if the items that Alice purchases, listens to or otherwise expresses interest in partially match the items of interest to Bob, then Alice and Bob share some tastes, and it is likely that Alice is also interested in the other items that interested Bob. Much like the AReXS application described in Chapter 5 decides that a pair of fields in an information source are semantically similar because their instances are syntactically similar, Del.icio.us infers similarity of personal interests by identifying similarities in people’s actions, specifically in the web sites that they choose to bookmark.

Digg allows people to nominate on-line resources as interesting, and then invites other users to vote for, or ‘digg’, the resource. The most popular resources are then presented for browsing, based on the assumption that people are interested in popular articles. Traditional news web sites are beginning to implement similar ‘popularity’ measures in their presentation of news stories; for example, Yahoo!7, a partnership between the Australian commercial television broadcaster Channel 7 and the on-line news, search and e-mail service provider Yahoo!, includes a ‘Most Viewed Stories’ panel alongside its news stories (see Figure 7.1). Similarly, the British Broadcasting Corporation’s (BBC) news web site invites readers to “[s]ee what’s popular and new” (Figure 7.2), and The Age newspaper encourages visitors to its web site to view “Today’s Top 10 Articles” as determined by the number of people who have read them (Figure 7.3).

The idea that popularity can identify quality is a problematic one, and diversity is important for revealing new and better ideas, but for the purposes of this proposed

---

2I am ignoring the space provided for discussion that accompanies every published page, as these are not part of the main body of Wikipedia and thus are unseen while reading an article — thus, only by actively seeking the corresponding discussion page does the reader become aware of any controversy surrounding a given entry. The removal of such arguments from the main entries to special ‘discussion pages’ clearly indicates that they are not considered equally privileged to the entries themselves.


CHAPTER 7. DATA DESIGN COLLABORATION

Figure 7.1: Web site of Yahoo7 showing ‘Most Viewed Stories’
TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS

Figure 7.2: Web site of the BBC showing ‘See what’s popular and new’
CHAPTER 7. DATA DESIGN COLLABORATION

**Figure 7.3:** Web site of The Age showing ‘Today’s Top 10 Articles’
tool for collaboratively reducing the problem of semantic interoperability, measuring
the number of applications or services that have adopted a particular data definition
allows developers to evaluate the potential benefits of adopting that data definition in
their own system.

Where Wikipedia attempts to capture agreed knowledge and Del.icio.us and Digg
capture common opinions, The Code Project and SourceForge are, as their names sug-
gest, focussed on source code and programming. SourceForge is a web-based project
hosting service that provides infrastructure to support collaborative open-source soft-
ware development. It particularly facilitates the formation and maintenance of develop-
ment teams over distances that would normally inhibit or prevent such co-operation. By
providing e-mail lists and discussion forums, bug and feature request tracking databases,
version control systems and download facilities for releases of applications, all with
commercial-grade reliability and availability, much of the house-keeping that would
otherwise be required to develop open-source software is taken care of and prospective
developers are left free to do what they do best. SourceForge hosts more than 150,000
projects, of which 7-10% are estimated to be active [26]. The Code Project, on the
other hand, can be traced back to on-line newsgroups and discussion groups that, since
the inception of the Internet, have provided programmers with forums to seek advice
on technical issues and to debate different styles and approaches to any software devel-
opment issue that they encounter. The willingness to freely give advice and assistance
surely derives form the same personality traits commonly found among computer pro-
grammers and software developers that lead to the open-source philosophy of sharing
and collaboration.

The Code Project collects on one web site a multitude of program fragments,
language-specific tips, implementation tutorials and debates over the relative merits
of various platforms, languages and frameworks. People are encouraged to post exam-
pies of implementing particular features or techniques, which other programmers are
welcome to adopt. Although there is no stated purpose of aligning implementations
of certain functions or achieving interoperability through common choices, there is a
strong element of collaboration that suggests that a site that encourages programmers
to share data definitions would be well received. It is clear that in the software develop-
ment industry, perhaps unlike other industries, there is a strong element of sharing of
work product and even what I would like to call ‘incidental collaboration’, where a pro-
grammer will respond to a request for advice and end up participating in a conversation
over days or even weeks until the original requester has solved their problem\textsuperscript{6}.

The success of both of these sites (and the others with similar goals that I cannot describe here) suggests that many people working in software development would be open to the idea of sharing their data definitions and schemas in the manner that I am suggesting.

The repository that is proposed, having been populated with data definitions for a wide variety of data objects and concepts, could then be accessed by any future system designer to discover representations that have been used for particular concepts or data. For example, designers for a new project that processes journeys might browse or search the repository and discover three applications that include a data object that represents a location that could be part of a journey. If the designers of the new application adopt this representation, they can reasonably believe that the opportunities for interoperation with the three existing applications are increased, as the effort necessary to translate data from their system to the form required by one of the existing systems would be significantly reduced, and vice versa. Similarly, one can imagine the possible savings of time and effort if the designers of the weather reporting, stock price quoting and on-line car advertising services presented in Chapter 6 had standardised their data formats prior to implementation.

### 7.2 Ontology repositories and the Semantic Web

Given the similarity between software system data object definitions and ontologies, both being explicit representations of conceptualisations, it is important to acknowledge that the idea of a published standard ontology or a repository of ontologies for reuse is not new. Along with the massive Cyc ontology described in Chapter 4, the Suggested Upper Merged Ontology (SUMO) was published in 2000 [101, 94] and the Standard Upper Ontology (SUO) was proposed around the same time by an IEEE working group [64]. Both of these efforts aimed to provide a standard set of high-level concepts from which more specific and detailed ontologies could inherit, the goal being that all the resulting ontologies would have in common their highest, and in some regards, most basic concepts, which would increase their compatibility and reduce the problem of semantic interoperability. However, neither effort gain traction among ontology creators. In [64], Legg notes the tendency of researchers and organisations to contribute existing, fully-formed ontologies, and it seems reasonable to imagine that

\textsuperscript{6}I think this is noteworthy because the problem in question might very well be part of the original requester’s paid work — it seems unlikely to me that many other industries could boast a significant number of professionals who would be willing to help a stranger do their job, for no reward themselves.
the difficulty of reconciling these discouraged the volunteer members, for whom their own ontologies were already serving their purposes well and who perhaps did not see sufficient benefits to be gained by re-designing their ontologies to fit in with others.

In Chapter 3, I discussed a number of projects that create, collect and publish domain-specific ontologies. Some of these efforts include manual reconciliation of the ontologies that they present. A variety of projects have come into existence with the goal of housing and publishing ontologies submitted by anyone who chooses to do so. For instance, the DARPA Agent Mark-up Language project published the DAML Ontology Library, which boasts 282 ontologies but doesn’t appear to have been updated since 2004, and indeed the DAML project as a whole ceased activity in 2006. It is unclear whether SchemaWeb, a project to collect RDF schemas and make them available to both people and software agents to browse and search, has seen any activity since 2005.

Swoogle is “a search engine for the Semantic Web on the Web” that crawls the Internet looking for RDF documents and schemas and then allows people to search for concepts among the many ontologies it has accumulated. It currently indexes over ten thousand ontologies. However, there is no obvious attempt to integrate the ontologies other than via relationships expressed within the ontologies themselves, such as the referencing, importing, extending and versioning provided for by the RDF. Swoogle has been criticised by [84] for the semantically scattered results that it returns, in that it doesn’t attempt to disambiguate search terms, nor does Regardless, there seems to be a significant energy behind the project, and it is possible that having the infrastructure to automatically identify, index and present any published ontology or schema could reduce the effort involved in manually reconciling ontologies sufficiently that people who otherwise might not have will now do so. More interestingly, a large, automatically updated repository might provide a good environment for large-scale automated alignment of ontologies, doing for ontologies what the system I am proposing in this chapter aims to do for data definitions.

It seems likely that at some point in the not-too-distant future there will be a convergence between information mark-up ontologies and software system data definitions. The two are so similar in purpose that it is difficult to believe that they will always remain separate. Increasingly, software developers are adopting XML-based representations for storing structured information, almost certainly due to its perceived

---

7 According to the project’s web site at http://www.daml.org/.
8 Based on the most recently-dated posts on the project’s web site at http://schemaweb.info/.
9 According to the project’s web site at http://swoogle.umbc.edu/.
portability, as well as the growing support for the format by development tools. The re-specifying of HTML as an XML document type [133] means that the infrastructure of the World Wide Web, as well as the tools that enable us to navigate, search and read its contents, further increases the incentive to store data in a form that can be readily expressed in XML, as style sheets make publishing such data easy, while retaining all its structure in a form that promises to increase interoperability. However, the semantic problem remains, and in this thesis I have described the many efforts to overcome this, none of which have yet been completely successful. With this in mind, I feel that there is an opportunity to encourage ontological consistency at the software data definition level. If the survey of efforts to date has shown anything, it is that when it comes to the goal of universal semantic interoperability, local, incremental solutions are necessary to advance the state of the art slowly toward that goal.

7.3 A prototype

With these issues in mind, a prototype a web site has been built that allows users to specify data object representations in an object-oriented style, as classes with data members, optionally extending other classes. Since it is imagined that the most likely use of the site would be that a software designer creates an initial proposal for the data structures required by their application, and then searches the repository for similar data structures, functions have been implemented to calculate the similarity of two classes based on their representation. This calculation is based on:

- the similarity of the class’ name, via string comparison algorithms similar to those used in ARexS; and
- the number of similarly-named data members, determined by string comparison of member names and by types of data members, such as \textit{int}, \textit{String}, et cetera.

Although not yet implemented, this function would almost certainly benefit from the inclusion of a more detailed exploration of the structure of the relationships between classes, including the existence of parents, siblings and children — this could be achieved by methods similar to those employed by the structural ontology alignment algorithms discussed in Chapter 3.

The initial view of the site is a list of projects that have been defined by the user, such as in Figure 7.4). Each project belongs to a person, so that the designer of a project can
particular data definition can be contacted if desired. The user can browse all of the projects defined in the system, as shown in Figure 7.5.

Selecting a project reveals that projects consist of data entities, expressed in an informal data dictionary format (see Figure 7.6. If desired, a simple UML-style class diagram can be generated to present the project’s data entities in a more visual form (Figure 7.7).

When viewing a project the user can request to see other classes similar to those defined for this project. Doing so results in a list being presented, containing every class defined in the repository that is in some way similar to the selected class, ordered by their similarity (Figure 7.8). In the example screen-shots, the class selected was a data entity that represents a film as screened by a cinema, and the top results by score appear to also be representations of films as defined by cinemas.

The final function of the site is to compare and contrast for the viewer their selected class and the classes from other projects that are claimed to be similar (Figure 7.9). The comparison page attempts to explain to the viewer how the two classes are similar by connecting similar attributes by lines, and also by colouring pairs of attributes that are themselves similar, to assist the viewer in visually identifying them. For example, in Figure 7.9, the attribute ‘Title’ of the left class is coloured red and so is the corresponding attribute ‘Title’ of the class on the right. However, the attribute ‘Title’ of the right class also matched weakly with the ‘Distributor’, ‘Trailer’ and ‘OfficialWebsite’ attributes of the class on the left, so these three attributes have been coloured with pale shades of pink to show their weak similarity without confusing the viewer. In the same way, the attribute ‘Rating’ of the right class matched strongly with the attribute of the same name on the left, which was coloured in strong green, but also with three other attributes on the left, each of which was given a light green shade. On the other hand, the ‘SessionTimes’ attribute of type ‘Time’ on the left matched moderately with the ‘SessionTimes’ attribute of type ‘Object’ on the right, but no other attributes, and so it is shaded with a moderately strong blue and no other attributes are coloured with any shade of blue.

### 7.4 Evaluation and conclusions

As the only way to evaluate a system based on the prototype presented in this chapter would be to measure it after a period of operation with a large number of users and projects, the scale of which would be well beyond that achievable in a thesis, I have not performed a technical evaluation. However, an informal exploration of the effectiveness
CHAPTER 7. DATA DESIGN COLLABORATION

Figure 7.4: My projects list in Data Design Collaboration web site
### All Projects

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton's Apple Project</td>
<td>What makes Newton so famous</td>
</tr>
<tr>
<td>Christmas Gifts Delivery System</td>
<td>The fastest and most reliable gift delivery system for this holiday</td>
</tr>
<tr>
<td>Flight Booking System</td>
<td>Book your flight tickets online</td>
</tr>
<tr>
<td>Hoyts Movie Site</td>
<td>Hoyts Cinemas</td>
</tr>
<tr>
<td>Village Movies Site</td>
<td>Village Cinemas</td>
</tr>
<tr>
<td>Nova Movies Site</td>
<td>Nova Cinemas</td>
</tr>
<tr>
<td>Google Calendar</td>
<td>Online Calendars</td>
</tr>
<tr>
<td>Calendars Net</td>
<td>Online Calendar</td>
</tr>
<tr>
<td>iCal</td>
<td>Mac Desktop Calendar</td>
</tr>
<tr>
<td>City Movies Site</td>
<td>City Cinemas</td>
</tr>
<tr>
<td>A Movie Website</td>
<td>Never exists</td>
</tr>
<tr>
<td>Another Movie Site</td>
<td>Never exists</td>
</tr>
<tr>
<td>My Calendar</td>
<td>my desktop calendar</td>
</tr>
<tr>
<td>Cool Calendar</td>
<td></td>
</tr>
</tbody>
</table>

**Compare Entity Names**: To test entity names, All attributes are the same

**Compare Attribute Names and Attribute Types**: Keep names of the classes the same

**Compare Entity Names and Attributes**: More than one attribute

#### Figure 7.5: All projects list in Data Design Collaboration web site
### Data Design Collaboration Project

#### Village Movies Site

- **URL**: www.villagecinemas.com.au
- **Description**: Village Cinemas
- **Tags**: cinema movie

#### Classes used in this project

<table>
<thead>
<tr>
<th>Classes</th>
<th>Attributes</th>
<th>Find Similar Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cinema</strong></td>
<td>State: String</td>
<td>Similar Classes</td>
</tr>
<tr>
<td></td>
<td>Location: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PhoneNo: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FaxNo: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NoOfScreens: Integer</td>
<td></td>
</tr>
<tr>
<td><strong>Movie</strong></td>
<td>Title: String</td>
<td>Similar Classes</td>
</tr>
<tr>
<td></td>
<td>Genre: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rating: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ConsumerAdvice: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ReleaseDate: Date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RunningTime: Integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainCast: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Director: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Distributor: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trailer: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OfficialWebsite: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SessionTimes: Time</td>
<td></td>
</tr>
<tr>
<td><strong>SessionTime</strong></td>
<td>Date: Date</td>
<td>Similar Classes</td>
</tr>
<tr>
<td></td>
<td>Time: Time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NFT: String</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7.6**: Project description in Data Design Collaboration web site
Figure 7.7: Class diagram for project in Data Design Collaboration web site
### Similar Classes

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Project Name</th>
<th>Rate</th>
<th>Class Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie</td>
<td>Village Movies Site</td>
<td>1</td>
<td>View</td>
</tr>
<tr>
<td>Movie</td>
<td>Hoyts Movie Site</td>
<td>0.92308974359</td>
<td>View</td>
</tr>
<tr>
<td>Movie</td>
<td>Nova Movies Site</td>
<td>0.820183760684</td>
<td>View</td>
</tr>
<tr>
<td>Movie</td>
<td>City Movies Site</td>
<td>0.757032245532</td>
<td>View</td>
</tr>
<tr>
<td>Movie</td>
<td>Another Movie Site</td>
<td>0.693645299145</td>
<td>View</td>
</tr>
<tr>
<td>Movie</td>
<td>A Movie Website</td>
<td>0.681978632479</td>
<td>View</td>
</tr>
<tr>
<td>CAR</td>
<td>Drive</td>
<td>0.171306915307</td>
<td>View</td>
</tr>
<tr>
<td>Event</td>
<td>Google Calendar</td>
<td>0.159714063714</td>
<td>View</td>
</tr>
<tr>
<td>PeriodicEvent</td>
<td>Calendars Net</td>
<td>0.154266511267</td>
<td>View</td>
</tr>
<tr>
<td>Cinema</td>
<td>Village Movies Site</td>
<td>0.144491841492</td>
<td>View</td>
</tr>
<tr>
<td>Calendar</td>
<td>Google Calendar</td>
<td>0.143529137529</td>
<td>View</td>
</tr>
<tr>
<td>Person</td>
<td>Compare Entity Names and Attributes</td>
<td>0.141825174825</td>
<td>View</td>
</tr>
<tr>
<td>Person</td>
<td>Compare Entity Names and Attributes</td>
<td>0.140397824398</td>
<td>View</td>
</tr>
<tr>
<td>Event</td>
<td>iCal</td>
<td>0.135863636364</td>
<td>View</td>
</tr>
<tr>
<td>Vehicle</td>
<td>AutoWeb</td>
<td>0.135567599068</td>
<td>View</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>Ebay</td>
<td>0.134047397047</td>
<td>View</td>
</tr>
<tr>
<td>Event</td>
<td>Calendars Net</td>
<td>0.133363636364</td>
<td>View</td>
</tr>
<tr>
<td>ToDo</td>
<td>iCal</td>
<td>0.132638306138</td>
<td>View</td>
</tr>
<tr>
<td>Car</td>
<td>Autotrader</td>
<td>0.132072649573</td>
<td>View</td>
</tr>
<tr>
<td>SessionTime</td>
<td>Village Movies Site</td>
<td>0.130363636364</td>
<td>View</td>
</tr>
<tr>
<td>HowToRepeat</td>
<td>Calendars Net</td>
<td>0.125333333333</td>
<td>View</td>
</tr>
<tr>
<td>Person</td>
<td>Compare Entity Names and Attributes</td>
<td>0.124377622378</td>
<td>View</td>
</tr>
<tr>
<td>Member</td>
<td>Compare Entity Names and Attributes</td>
<td>0.124377622378</td>
<td>View</td>
</tr>
<tr>
<td>SessionTime</td>
<td>A Movie Website</td>
<td>0.124325174825</td>
<td>View</td>
</tr>
</tbody>
</table>

**Figure 7.8:** Similar classes list in Data Design Collaboration web site
**Figure 7.9:** Comparing two similar classes in Data Design Collaboration web site
of the principle by which it is motivated has been conducted. In order to populate
the repository with data that might be representative of real development projects, a
number of cinema web sites were analysed to deduce the data definitions that were
apparently adopted by the sites’ developers, and a number of calendar applications
were similarly analysed to identify likely data definitions chosen by developers, based
on the presentation of this data through the applications’ user interface. Each web site
and application resulted in several class definitions, which were added to the repository.
The resulting (small) collection of projects and classes was then explored by selecting
classes arbitrarily and requesting that similar classes be displayed — in no case were
the results unexpected.

As with all systems whose usefulness increases with the number of interconnected
users or content, the only way to verify the value that I hope would result from large-
scale adoption of the system I am proposing would be to create a large version of the
system with many users. Obviously, this is infeasible for a small research project, as it is
difficult to predict how much time such an experiment would require. Despite this, I do
believe that the potential of this proposal is supported by the successes of projects with
a similar basis, as described earlier in this chapter. A further encouragement comes
from the Microformats project, decentralised attempt to establish a set of agreed-upon
formats for common data objects, such as “contact information, events, reviews, episodic
content, etc” [86] (see [1] for more information about microformats). Efforts such as
these are arguably the way that a semantic web will actually eventuate. As discussed
in Chapter 3, the Semantic Web project has so far provided a promising infrastructure
for semantic interoperation but hasn’t yet offered a convincing solution to the problem
of agreeing on the meanings of shared terms.

Even after adopting the RDF or OWL representations, it remains necessary to seek
out existing ontology terms, concepts and relationships in order to achieve interoper-
ability. The World Wide Web did not become the critically useful system that it is
today in a single step. Indeed, referring to the World Wide Web as a single system
misses an essential aspect of it’s nature: it is at heart a collection of millions of smaller
systems or communities that are constructed by and useful to generally small (although
in some cases huge) groups of people. The World Wide Web, as we know it today, was
not planned or designed; instead, it grew from a multitude of smaller efforts, and syner-

12Apple’s iCal, Google Calendar, Calendars.Net
13A phenomenon commonly called Metcalfe’s Law after Robert Metcalfe who used the idea in the
early 1980’s to convince businesses to purchase Ethernet networks.

161
gies and opportunities emerged. In the same way, any semantic web that we might end up with will come from a sufficient combination of user desire and ease of use. When there is demand from enough people, standards tend to emerge; whether they are “official” jointly-agreed standards such as JPEG, MP3 and HTML, or proprietary de facto standards such as Microsoft’s Word document format, there are many examples from recent history alone that demonstrate a tendency for people to agree on definitions and protocols in order to communicate and interact. To date, the requirement of learning how to write in RDF and the labouriousness of tagging every word and object with a URI have outweighed the level of interest in the promised benefits of a semantic web. I believe that it will be through many minor enabling tools and technologies such the data design collaboration site described in this chapter that a world-wide network of semantic interoperability will be achieved.

As a final comment, I argued earlier in this chapter that local, incremental efforts are necessary to progress toward the goal of universal interoperability. Tagging and folksonomy [83] of the nature employed by some of the projects described earlier such as Del.icio.us is one such incremental effort. Folksonomy, or the authority of the expressed perspectives of individuals over pre-defined, imposed ontologies, has recently become the subject of some discussion [132, 111]. It is arguably a re-visiting of the idea of ethnoclassification [85], such as discussed by Star in 1996 [121], which was, as the name suggests, an application of ethnographic principles to the creation of classification schemes for the artefacts that people encounter and produce in the course of their work. As evidence of the value perceived by some, it has been posited that the user-generated tags that have been applied to the Flickr library of images14 was one of the factors that lead the Internet search and directory company Yahoo! to buy the library, the argument being that such tags are the best hope for enabling textual searching of images [99].

It is difficult to see yet an exact path from ad hoc tags to ontologically sound markings. As an example, searching at the Deli.icio.us site for items tagged ‘interesting’ finds approximately 215,000 links, but, given that it is a site for bookmarks, which are either pages that people want to read or re-read or pages that they would like to share with others, I can’t help but wonder why any items are not considered interesting. Similarly, querying the Flickr site for images tagged “me” currently results in nearly 2.2 million photographs which are obviously not all pictures of the same person.

Examples like this cause immediate concern to proponents of ontologies and taxonomies, for obvious reasons: what if the tags that people choose to apply don’t form a coherent vocabulary? Others, such as Shirky, argue that no prescribed categorisation

---

scheme can truly accommodate users’ needs as adequately as we would like [112]. I would add, it doesn’t seem necessary to include all tags in any effort to take advantage of their classifying ability. Simple heuristics can be employed to quickly exclude tags that are considered to be problematic or to parse only those that are predicted to contribute positively to whatever task is at hand, in a manner similar to that demonstrated by the SportsReporter software agent described in Chapter 4.

Shirky claims that user-generated informal tags should be viewed through the same perspective from which they were created: that they mean something, to someone, but that others are free to take them or leave them as they see fit. In Shirky’s view, if the terms ‘film’, ‘cinema’ and ‘movie’ are all synonymous to you, by all means equate them, but allow for the possibility that to other people, they are not [112]. This comes back to the argument with which I began this thesis: that different people, in different situations, will require different understandings of the portion of the world that they are dealing with at that time, and that software systems that are able to accommodate this in some way will be more useful than ones that require adoption of a prescribed view. I believe that local actions and implementations will slowly reveal ways that universal goals such as the Semantic Web can be realised, where top-down solutions will inevitably fail.

Publishing data definitions and providing means to quickly and easily find and compare other people’s decisions for representing date similar to yours will not solve the problem of semantic interoperability, but I think that it could provide another step toward the goal. Along these lines, I find it interesting to note that users of the Flickr photograph sharing web site have begun encoding what are effectively data triples (much like those on which the RDF data model is based) into the technically free-text tags applied to images [17]. As announced by [20], “machine tags” are simply an informal standard honoured by the Flickr programming interface for embedding ‘namespace’, ‘predicate’, ‘value’ triples into ordinary image tags, with the intention that they will typically be generated and processed by software applications. Examples given include “geo:locality=san francisco” and “flora:tree=coniferous”, as well as suggestions for referring (implicitly) to existing ontologies such as the Dublin Core Metadata Element Set [31] by “dc:title=...” and so on. No ontology is being imposed, nor even a reasoning framework, and yet it seems impossible that this will not be widely adopted and put to effective use.

It seems to me that if it is to be successful, the Semantic Web will grow as the original World Wide Web did, incrementally and in a widely distributed fashion, with occasional confluences and synergies and only eventually perhaps a universal breakthrough such
as that brought to the Web by Google’s algorithm for search. The usefulness of the
Page Rank algorithm was not apparent until the nature of the World Wide Web was
evident through the ways that its users adapted it to their many needs; likewise, I would
argue that approaches such as the data definition collaboration project presented in this
chapter are necessary stepping stones on the path to semantic interoperability.
Chapter 8

Conclusion

This thesis has travelled across several aspects of the current effort toward greater integration, interaction and interoperation of our information systems. The recent history of information integration has been presented, beginning with logic programs superimposed on relational databases permitting reasoning about their fields and records, passing through ontology alignment techniques and semantic mark-up technologies, and then delving into three approaches to facilitating semantic interoperability that are quite different in philosophy, but closely related in their objectives.

I began this thesis by defining the problem of semantic interoperability and exploring its origins in the nature of software as a symbolic model of a set of concepts. Computer programs, database schema and knowledge representations are all models of ideas, concepts and relationships, and for any idea there are many different ways to express it as a symbolic model. When people design software systems to produce and process information, they make choices about how to represent this information within the boundaries of the framework or architecture that they have chosen. They also make more arbitrary decisions that are not the result of system constraints, but merely the result of personal, traditional and conventional experiences and beliefs. The result of these decisions is that the models designed diverge in their representation of similar or equivalent concepts and relationships, leaving the implemented programs and systems incompatible.

As presented in Chapter 3, the past several decades have seen a variety of approaches to countering this divergence and providing means to reconcile or align systems that manipulate equivalent concepts but are not capable of interoperating directly. None of these strategies have so far completely resolved the problem, and so this thesis explored three different approaches to the problem, each of which is either new, or derived from
a little-explored previous work. While none of the paths explored in this thesis lead to a complete solution, it is my belief that progress has been made and that the results described herein will be of value to the other work currently being performed in this field. The more that it becomes clear that hybrid, heuristic approaches are necessary to deal with the complexities of the problem of semantic interoperability, the more valuable each individual approach or technique becomes as a complement to other strategies — whereas complete solutions compete to solve problems, heuristic solutions tend by contrast to be co-operative and collaborative, and architectures that acknowledge the fragmented, semantically divergent nature of knowledge representation and processing invite techniques that contribute hints and suggestions to an overall attempt to tackle problems that may have no actual solution, but merely degrees of useful results.

8.1 Experimental results

The specific research question that was addressed in this thesis was: given the previous, only partially successful, attempts at automatically reconciling the ontological perspectives of different software applications and agents, what different approaches can be made and in what situations can they expect to be successful? To answer this question, three experiments were devised and performed, as described in Chapters 4, 5, 6 and 7. Based on real-world situations and where possible real data, each experiment identified an avenue of approach to solving the problem of semantic interoperability.

The SportsReporter information agent presented in Chapter 4 was built to integrate information found on web pages into the Cyc knowledge-base, extending the Structured Knowledge Source Integration project to include semi-structured and unstructured information sources. I was able to demonstrate the generalisation of the knowledge required to build a lightweight information retrieval agent into a form that can itself be expressed within Cyc, paving the way for Cyc to generate new agents as required to retrieve information that it doesn’t yet possess. Although this goal was not achieved, I believe that the work presented in this thesis has provided the foundation for future progress.

The Cyc project has survived for over twenty years, building a massive formal knowledge-base that contains literally millions of common sense rules and relations. One area of weakness that has been identified by the Cyc developers themselves is the problem of incorporating large amounts of raw data that serve as instances of many of its concepts. Although an internal Cyc project aims to address this by describing structured databases in such a way that Cyc can attach meaning to their tables and
CHAPTER 8. CONCLUSION

fields and thus reason about their contents, I have described the process of integrating with Cyc a lightweight, heuristic-driven software agent that retrieves information from semi- or unstructured sources such as web pages. Further, I removed the task-specific knowledge from the information agent and incorporated it into Cyc itself, thereby generalising the agent in order to provide Cyc with the ability to generate other, similar agents as necessary to retrieve the information required to answer future questions.

The AReXS application presented in Chapters 5 and 6, derived from the EBFM algorithm published in 1998 [56] and subsequently ignored to the best of my knowledge, was quite successful in automatically reconciling semantically equivalent fields from databases with incompatible schema without any domain knowledge. Research into ontology alignment techniques has been criticised for being evaluated only on artificial data and for not sufficiently explaining the assumptions behind the evaluations [8]. For this reason, I have provided extensive details of the data used in this experiment, including its source, how it was processed prior to use, and how the results of the experiment were interpreted for evaluation. I believe that the results of this experiment show that the strengths of this approach presented in Chapter 5 lie in the very areas in which other ontology alignment algorithms are weakest, namely where concept or field labels are missing, incomplete or otherwise not easily equated by the techniques commonly employed in the field of ontology alignment. Evaluating the algorithm’s performance via the standard measures of precision and recall lead to results that I contend would generally be considered useful.

Instance-based methods can demonstrably serve as an effective tool for reconciling information sources that have been created with divergent ontological perspectives. Often, as presented in Chapter 3, data is accompanied by structured, formal ontologies and thus the many ontology alignment algorithms that focus on structural characteristics can be employed. But in many cases this level of structure is not available, and I have presented examples of these cases. Further, the final data set to which the AReXS application was applied was deliberately messy, chosen as an example that would test the limits and probe the weaknesses of instance-based approaches. Based on analysis of the results of this experiment, I identified and described a set of data types that are not handled well by the standard string-oriented instance comparison techniques employed by all the instance-based matching approaches identified in Chapter 3, and where possible I proposed alternative techniques.

The data design collaboration project presented in Chapter 7 is a novel approach to the problem of semantic interoperability, proposing that the differences of representation that arise during data design be addressed prior to implementation rather than
after a system has been deployed. The reasoning behind this proposal and prototype is derived from the software engineering principle that faults that require changes to a system’s behaviour increase by orders of magnitude as the development project moves from phase to phase, and therefore the earlier a required change can be identified and made, the less effort it will require.

The problem of communication and consensus among the developers of different systems that are likely to have been designed at different times and in disparate places is addressed by applying the social aspects of successful collaborative projects such as Wikipedia, Sourceforge, Del.icio.us and digg, as well as the Semantic Web approach of publishing domain-specific ontologies in the hope of encouraging others to share and adopt their representations. Some of the similarity identification techniques employed by the AReXS algorithm were applied to assisted developers in locating data representations similar to their own, at which point they have the opportunity to modify their representations to be more compatible with existing systems, thus reducing the barriers to interoperation between systems. Although only informal evaluation was performed, I believe that the principles on which the project is based are sound, and that the growing significance of similar efforts such as the Microformats project lends support to this belief.

8.2 Future work

As I have identified, the problem of semantic interoperability is an extremely complex one that is far from solved. Each of the works presented in the body of this thesis is in some respect incomplete, in that they neither set out to nor did completely solve the problems that they address. For each, however, I see certain activities that would be sensible and, I believe, valuable “next steps” on the path to the ultimate goal of semantic interoperability for software systems.

The results of the integration of the SportsReporter information agent with the Cyc knowledge-base show that there is clear potential for automatically generating task-specific agents to retrieve information as required. In Chapter 4 I presented a framework that is general enough to be applied to a large class of “question answering” tasks. A larger scale evaluation, and the identification of other classes of questions that could be answered for the Cyc knowledge-base in a similar fashion, would further strengthen this work. Although a large-scale formal evaluation was beyond the scope of this thesis, I believe that it would help to demonstrate the versatility of the approach presented and to identify the limits of the approach, both of which are important
The experiment in automatically reconciling semi-structured heterogeneous information sources with the AReXS application showed that there are effective solutions to an area of the problem of semantic interoperability that is under-explored in the field of ontology alignment. It seems clear now after a decade of strong research in ontology alignment that further progress will require blended approaches that combine structural, linguistic and instance-based algorithms, as well as heuristics for data type recognition and identification. I believe that the results achieved in Chapter 6 demonstrate that the work presented in this thesis complements other research in the field.

In order to properly evaluate the data design collaboration project presented in Chapter 7 it will be necessary to conduct a large-scale experiment. One reasonable approach would be to build and launch a public version of the web site and encourage software developers to use it, although such an experiment would contain many variables and it would be difficult to determine success for some time. Another approach would be to retrieve a large number of previously developed software projects and extract their data designs to seed the collaboration system, after which experiments could be defined to evaluate the potential for interoperability that could be identified. Public repositories of open source projects are obvious candidates for such an experiment, but I believe that considering the collaboration system as a means for increasing intra-organisational interoperability could provide benefits by identifying internal projects that process similar data but have not adopted consistent schemas. However, to do this would require the co-operation of one or more large companies or institutions.

This thesis has considered in depth the problem of semantic interoperability for software systems, analysed its origins and consequences and identified, proposed and where possible evaluated approaches to overcome the problem. Although the problem in general remains unsolved, I believe that this thesis has advanced the state of the art in the ways described above.
References


TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS


REFERENCES


[21] Corcho, O. & A. Gómez-Pérez, Solving Integration Problems of E-Commerce Standards and Initiatives through Ontological Mappings in Asuncion Gómez-Pérez, Michael Grüninger, Heiner Stuckenschmidt & Michael Uschold (eds), Workshop on Ontologies and Information Sharing, 17th International Joint Conference on Artificial Intelligence (IJCAI ’01), Seattle, 2001, pp 131–140


the 2001 ACM SIGMOD International Conference on Management of Data (SIGMOD '01), Santa Barbara, 2001


[34] Engmann, Daniel & Sabine Maßmann, Instance Matching with COMA++, in Proceedings of the Workshop on Datenbanksysteme in Business, Technologie und Web (BTW '07), Aachen, 2007


[38] Faatz, Andreas & Ralf Steinmetz, Precision and Recall for Ontology Enrichment, in Proceedings of the Workshop on Ontology Learning and Population at the 16th European Conference on Artificial Intelligence (ECAI ’04), Valencia, 2004


[51] Heflin, Jeff & Hendler, James, **Dynamic Ontologies on the Web** in *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI '00)*, AAAI/MIT Press, Menlo Park, 2000


[59] Kim, Wooju, Sangan Park, Siri Bang & Sungwhan Lee, **An Ontology Mapping Algorithm between Heterogeneous Product Classification**
REFERENCES


[63] Legg, Catherine, Implementation of a Large-scale General Ontology at Cycorp, Departmental seminar, Department of Computer Science and Software Engineering, The University of Melbourne, Melbourne, 2002


[68] Lister, Kendall & Leon Sterling, Reconciling Ontological Differences for Intelligent Agents, in Paolo Bouquet (ed.), Proceedings of Meaning Negotiation,


[70] Lister, Kendall & Leon Sterling, Reconciling Heterogeneous Information Sources, in Proceedings of the 3rd International Semantic Web Conference (ISWC '04), Hiroshima, 2004


[74] Lu, Hongen, Leon Sterling & Alex Wyatt, Knowledge Discovery in SportsFinder: An Agent to Extract Sports Results from the Web, in Ning Zhong & Lizhu Zhou (eds), Methodologies for Knowledge Discovery and Data Mining, Proceedings of the 3rd Asia-Pacific Conference (PAKDD '99), Beijing, China, Springer-Verlag LNAI, 1999, pp 469–473


[82] Masters, James, Structured Knowledge Source Integration and its applications to information fusion, in Proceedings of the 5th International Conference on Information Fusion, Annapolis, 2002


TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS


[93] Naumann, Felix, Ching-Tien Ho, Xuqing Tian, Laura M. Hass & Nimrod Megiddo, *Attribute classification using feature analysis*, in *Proceedings of the 18th International Conference on Data Engineering (ICDE ’02)*, San Jose, 2002


[103] Rahm, Erhard, Hong-Hai Do & Sabine Maßmann, **Matching Large XML Schemas**, *Sigmod Record*, Vol. 33, No. 4, 2004

[104] Rijsbegeur, C. J. van, **Information Retrieval** Butterworths, London, 1979


[106] Schaaf, Martin, & Kudger van Elst, **An Approach To Cooperating Organizational Memories Based On Semantic Negotiation and Unification**, in *Proceedings of the Workshop on Meaning Negotiation at the 18th National Conference on Artificial Intelligence (AAAI ’02)*, Edmonton, 2002


REFERENCES


TOWARD SEMANTIC INTEROPERABILITY FOR SOFTWARE SYSTEMS

[128] Sycara, Katia, **Multi-agent Infrastructure, Agent Discovery, Middle Agents for Web Services and Interoperation, Multi-Agent Systems and Applications**, in *Proceedings of 9th ECAI Advanced Course on Artificial Intelligence (ACAI ’01)*, Prague, 2001


[130] Theodosiev, T., **Multi-Agent System with Sociality**, in *Proceedings of Adaptability and Embodiment Using Multi-Agent Systems, Advanced Course on Artificial Intelligence (ACAI ’01)*, Prague, 2001


[137] Wu, Shengli, Qili Zhou, Yaxin Bi & Ziaqin Zeng, **Performance Weights for the Linear Combination Data Fusion Method in Information Retrieval**, 184
REFERENCES


APPENDIX A

The following pages contain the survey that was used to derive the set of results used as the correct standard for the AReXS experiment presented in Chapter 6.
Matching information sources

Instructions:

Cars offered for sale on Australian web sites have been recorded and are presented on the following pages.

Please examine the car descriptions presented and decide which fields from each column you consider to represent the same information. For example, on the second page, Autotrader vs AutoWeb, you might decide that the STATE field from the first column matches the LOCATION field from the second column.

Please then write these two field names in the table at the bottom of the page, and indicate how confident you are that the two fields represent the same information. For example, you might be very confident that the STATE and LOCATION fields represent the same information, and so you might write 100%. If you decide that there is some similarity between the two fields, but you are not very confident that they represent the same information, you might write 50%.

If it helps you, you could consider that you have been asked to join the two databases presented and you must decide which fields you can merge and which you cannot.
<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Mazda 323 Astina</th>
<th>YR: 1992</th>
<th>KMS: 140,000</th>
<th>PRICE: $9,999</th>
<th>STATE: QLD</th>
<th>DEALER: Ferrierr Qualit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR</td>
<td>MAKE_MODEL: Ford Falcon XH</td>
<td>YR: 1997</td>
<td>KMS:</td>
<td>PRICE: $9,999</td>
<td>STATE: VIC</td>
<td>DEALER: Lucky Choice Ca</td>
</tr>
<tr>
<td>CAR</td>
<td>MAKE_MODEL: Mitsubishi Magna</td>
<td>YR: 1994</td>
<td>KMS: 142,000</td>
<td>PRICE: $8,990</td>
<td>STATE: NSW</td>
<td>DEALER: Laurieton Motor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Autotrader</th>
<th>Autotrader</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Autotrader vs Autotrader
<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Mazda 323 Astina</th>
<th>YR: 1992</th>
<th>KMS: 140,000</th>
<th>PRICE: $9,999</th>
<th>STATE: QLD</th>
<th>DEALER: Ferrierr Qualit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR</td>
<td>MAKE_MODEL: Ford Falcon XH</td>
<td>YR: 1997</td>
<td>KMS:</td>
<td>PRICE: $9,999</td>
<td>STATE: VIC</td>
<td>DEALER: Lucky Choice Ca</td>
</tr>
<tr>
<td>CAR</td>
<td>MAKE_MODEL: Ford Fairlane</td>
<td>YR: 1991</td>
<td>KMS: 176,330</td>
<td>PRICE: $9,000</td>
<td>STATE: NSW</td>
<td>DEALER:</td>
</tr>
<tr>
<td>CAR</td>
<td>MAKE_MODEL: Mitsubishi Magna</td>
<td>YR: 1994</td>
<td>KMS: 142,000</td>
<td>PRICE: $8,990</td>
<td>STATE: NSW</td>
<td>DEALER: Laurieten Motor</td>
</tr>
</tbody>
</table>

| VEHICLE   | YEAR: 1992 | DESCRIPTION: MITSUBISHI MAGNA | PRICE: $9,990 |
|           |           | BODY_TYPE: SEDAN | COLOUR: WHITE/GREY | LOCATION: HAWTHORN, VIC |

| VEHICLE   | YEAR: 1996 | DESCRIPTION: FORD FALCON | PRICE: $9,200 |
|           |           | BODY_TYPE: WAGON | COLOUR: | LOCATION: NEWCASTLE AND CENTRAL |

| VEHICLE   | YEAR: 1995 | DESCRIPTION: HOLDEN BARINA SWING | PRICE: $6,995 |
|           |           | BODY_TYPE: | COLOUR: | LOCATION: NEWCASTLE AND CENTRAL |

| VEHICLE   | YEAR: 1994 | DESCRIPTION: SUBARU SPORTSWAGON | PRICE: $7,990 |
|           |           | BODY_TYPE: COLOUR: | LOCATION: SYDNEY |

| VEHICLE   | YEAR: 1996 | DESCRIPTION: MITSUBISHI LANCER GLI CE | PRICE: $9,990 |
|           |           | BODY_TYPE: COUPE | COLOUR: METALLIC BLUE | LOCATION: |

| VEHICLE   | YEAR: 1994 | DESCRIPTION: MITSUBISHI MAGNA TS | PRICE: $7,995 |
|           |           | BODY_TYPE: SEDAN | COLOUR: MAROON | LOCATION: MT. GAMBIER, SA |

<table>
<thead>
<tr>
<th>Autotrader</th>
<th>AutoWeb</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Autotrader vs CarSales

<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Mazda 323 Astina</th>
</tr>
</thead>
<tbody>
<tr>
<td>YR:</td>
<td>1992</td>
</tr>
<tr>
<td>KMS:</td>
<td>140,000</td>
</tr>
<tr>
<td>PRICE:</td>
<td>$9,999</td>
</tr>
<tr>
<td>STATE:</td>
<td>QLD</td>
</tr>
<tr>
<td>DEALER:</td>
<td>Ferrierr Qualit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Saab 9000</th>
</tr>
</thead>
<tbody>
<tr>
<td>YR:</td>
<td>1987</td>
</tr>
<tr>
<td>KMS:</td>
<td>$5,990</td>
</tr>
<tr>
<td>STATE:</td>
<td>SA</td>
</tr>
<tr>
<td>DEALER:</td>
<td>Auto Credit Co</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Subaru Liberty GX</th>
</tr>
</thead>
<tbody>
<tr>
<td>YR:</td>
<td>1993</td>
</tr>
<tr>
<td>KMS:</td>
<td>$9,990</td>
</tr>
<tr>
<td>STATE:</td>
<td>VIC</td>
</tr>
<tr>
<td>DEALER:</td>
<td>Eastern Vehicle</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Ford Falcon XH</th>
</tr>
</thead>
<tbody>
<tr>
<td>YR:</td>
<td>1997</td>
</tr>
<tr>
<td>KMS:</td>
<td>$9,999</td>
</tr>
<tr>
<td>STATE:</td>
<td>VIC</td>
</tr>
<tr>
<td>DEALER:</td>
<td>Lucky Choice Ca</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Ford Fairlane</th>
</tr>
</thead>
<tbody>
<tr>
<td>YR:</td>
<td>1991</td>
</tr>
<tr>
<td>KMS:</td>
<td>176,330</td>
</tr>
<tr>
<td>PRICE:</td>
<td>$9,000</td>
</tr>
<tr>
<td>STATE:</td>
<td>NSW</td>
</tr>
<tr>
<td>DEALER:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Mitsubishi Magna</th>
</tr>
</thead>
<tbody>
<tr>
<td>YR:</td>
<td>1994</td>
</tr>
<tr>
<td>KMS:</td>
<td>142,000</td>
</tr>
<tr>
<td>PRICE:</td>
<td>$8,990</td>
</tr>
<tr>
<td>STATE:</td>
<td>NSW</td>
</tr>
<tr>
<td>DEALER:</td>
<td>Laurieton Motor</td>
</tr>
</tbody>
</table>

### Confidence (%)

<table>
<thead>
<tr>
<th>Autotrader</th>
<th>CarSales</th>
<th>Unknown1: 1986 MAZDA RX7 FC1031 Limited, 2 door COUPE, white, 4 sp Automatic, 170000km , Petrol Rotary 1.3 2cyl (1308cc). Reg. No UDK-569</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Unknown2: $6,500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown3: Only $6500 ONO. price is negotiable. the 5th picture is what the car COULD look like.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown4: Fulham Gardens, Adelaide, SA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown1: 1993 FORD LASER KH S, 5 door HATCHBACK, green, 5 sp Manual, 245000km , Petrol Multi-point injected 1.8 4cyl (1840cc). Reg. No NBS 437</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown2: $6,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown3: laser 93KH sports,1.8 EFI, air cond, p-steer,t-windows, $6000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown4: Wodonga, VIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown1: 1997 PROTON SATRIA GLi, 3 door HATCHBACK, GREEN, 5 sp Manual, 112000km , Petrol Multi-point injected 1.5 4cyl (1468cc). Reg. No UNH-993</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown2: $5,950</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown3: Good clean economical car - priced to sell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown4: Castle Hill, NSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown1: 1984 MERCEDES 230E W123 , 4 door SEDAN, Champagne (gold), 4 sp Automatic, 220000km , Petrol Multi-point injected 2.3 4cyl (2299cc). Reg. No OIL363</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown2: $6,500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown3:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown4: Brighton, VIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown1: 1992 FORD FALCON EB II GLi, 4 door WAGON, red, 4 sp Automatic, 200000km , Petrol Multi-point injected 4.0 6cyl (3984cc). Reg. No 415GTC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown2: $5,100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown3: 6 seater</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown4: Warner Brisbane, QLD</td>
</tr>
</tbody>
</table>
### Autotrader vs Drive

<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Mazda 323 Astina</th>
<th>MAKE_MODEL: Saab 9000</th>
<th>MAKE_MODEL: Subaru Liberty GX</th>
<th>MAKE_MODEL: Ford Falcon XH</th>
<th>MAKE_MODEL: Ford Fairlane</th>
<th>MAKE_MODEL: Mitsubishi Magna</th>
</tr>
</thead>
<tbody>
<tr>
<td>KMS:</td>
<td>140,000</td>
<td></td>
<td>142,000</td>
<td>176,330</td>
<td>175,000</td>
<td>142,000</td>
</tr>
<tr>
<td>PRICE:</td>
<td>$9,999</td>
<td>$5,990</td>
<td>$9,990</td>
<td>$9,999</td>
<td>$9,000</td>
<td>$8,990</td>
</tr>
<tr>
<td>STATE:</td>
<td>QLD</td>
<td>SA</td>
<td>VIC</td>
<td>VIC</td>
<td>NSW</td>
<td>NSW</td>
</tr>
<tr>
<td>DEALER:</td>
<td>Ferrierr Qualit</td>
<td>Auto Credit Co</td>
<td>Eastern Vehicle</td>
<td>Lucky Choice Ca</td>
<td></td>
<td>Laurieton Motor</td>
</tr>
</tbody>
</table>

### Unknown1: MERCEDES-BENZ 380
- **year:** 0
- **kms:**
- **trans:**
- **body:**
- **colour:**
- **ad_date:** 03/05/2003
- **features:** Anti lock braking, Alarm system, Cruise control, Radio cassette...
- **description:** MERCEDES 380SEL, Diamond Blue, blue lther, recond motor, Alpine stereo, alarm, ABS, sunrf, cruise, reg 1/04 YDQ756. $9500 ono 0439 311 767.
- **location:** The Sydney Morning Herald

### Unknown2: HOLDEN COMMODORE
- **year:** 1993
- **kms:** 175000
- **trans:** A
- **body:**
- **colour:**
- **ad_date:** 03/05/2003
- **features:** Power steering, Air conditioning...
- **description:** COMMODORE VP, '93 Series 11, auto., air, steer, $5200 ono. YNR-126. PH. 4959 1992
- **location:** The Newcastle Herald and Post

### Unknown1: HOLDEN COMMODORE
- **year:** 1992
- **kms:**
- **trans:**
- **body:**
- **colour:**
- **ad_date:** 03/05/2003
- **features:**
- **description:** COMMODORE VP 1992 Series II, TSU-387, regd Nov '03, new paint, tyres, very good cond, 175,000 ks, $7250 ono. 4965 9112.
- **location:** The Newcastle Herald and Post

### Unknown2: TOYOTA CAMRY
- **year:** 1991
- **kms:** 53000
- **trans:** A
- **body:**
- **colour:**
- **ad_date:** 03/05/2003
- **features:**
- **description:** TOYOTA CAMRY '91 EXECUTIVE AUTO ONE OWNER 53,000 K's $6500. RWK-150. PH: 4946 5548
- **location:** The Newcastle Herald and Post

### Autotrader Drive Confidence (%)

<table>
<thead>
<tr>
<th>Autotrader</th>
<th>Drive</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Autotrader vs eBay

<table>
<thead>
<tr>
<th>CAR</th>
<th>MAKE_MODEL: Mazda 323 Astina</th>
<th>YR: 1992</th>
<th>KMS: 140,000</th>
<th>PRICE: $9,999</th>
<th>STATE: QLD</th>
<th>DEALER: Ferrierr Qualit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR</td>
<td>MAKE_MODEL: Ford Falcon XH</td>
<td>YR: 1997</td>
<td>KMS:</td>
<td>PRICE: $9,999</td>
<td>STATE: VIC</td>
<td>DEALER: Lucky Choice Ca</td>
</tr>
<tr>
<td>CAR</td>
<td>MAKE_MODEL: Ford Fairlane</td>
<td>YR: 1991</td>
<td>KMS: 176,330</td>
<td>PRICE: $9,000</td>
<td>STATE: NSW</td>
<td>DEALER:</td>
</tr>
<tr>
<td>CAR</td>
<td>MAKE_MODEL: Mitsubishi Magna</td>
<td>YR: 1994</td>
<td>KMS: 142,000</td>
<td>PRICE: $8,990</td>
<td>STATE: NSW</td>
<td>DEALER: Laurieton Motor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VEHICLE</th>
<th>BIDS:</th>
<th>CAR:</th>
<th>PRICE: AU $6,000.00</th>
<th>TIME_LEFT: 4d 02h 33m</th>
</tr>
</thead>
<tbody>
<tr>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR:</td>
<td>MITSUBISHI MAGNA TJ SPORTS 2001 MODEL</td>
<td>PRICE: AU $25,000.00</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 5</td>
<td>CAR: MUST BE SEEN. SERIES II VR SS V8 5 SPEED</td>
<td>PRICE: AU $12,100.00</td>
<td>TIME_LEFT: 17h 30m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: BMW E30 Series John Player Special 323i Coupe</td>
<td>PRICE: AU $10,950.00</td>
<td>TIME_LEFT: 9d 16h 51m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 10</td>
<td>CAR: 1985 Holden VK Commodore Executive Unreg</td>
<td>PRICE: AU $500.00</td>
<td>TIME_LEFT: 6d 14h 12m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 1</td>
<td>CAR: Mitsubishi L300 van</td>
<td>PRICE: AU $1,000.00</td>
<td>TIME_LEFT: 2d 01h 14m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: 69 Camaro 396 RS/SS Matching # car</td>
<td>PRICE: AU $40,000.00</td>
<td>TIME_LEFT: 5d 00h 29m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: fuel saver economy increases lowers emissions</td>
<td>PRICE: AU $35.00</td>
<td>TIME_LEFT: 6d 01h 56m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Autotrader</th>
<th>eBay</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AutoWeb</td>
<td>Autotrader</td>
<td>Confidence (%)</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td><strong>VEHICLE</strong></td>
<td><strong>CAR</strong></td>
<td></td>
</tr>
<tr>
<td>YEAR: 1992</td>
<td>MAKE_MODEL: Mazda 323 Astina</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION: MITSUBISHI MAGNA</td>
<td>YR: 1992</td>
<td></td>
</tr>
<tr>
<td>PRICE: $9,990</td>
<td>KMS: 140,000</td>
<td></td>
</tr>
<tr>
<td>BODY_TYPE: SEDAN</td>
<td>PRICE: $9,999</td>
<td></td>
</tr>
<tr>
<td>COLOUR: WHITE/GREY</td>
<td>STATE: QLD</td>
<td></td>
</tr>
<tr>
<td>LOCATION: HAWTHORN, VIC</td>
<td>DEALER: Ferrierr Qualit</td>
<td></td>
</tr>
<tr>
<td><strong>VEHICLE</strong></td>
<td><strong>CAR</strong></td>
<td></td>
</tr>
<tr>
<td>YEAR: 1996</td>
<td>MAKE_MODEL: Saab 900</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION: FORD FALCON</td>
<td>YR: 1987</td>
<td></td>
</tr>
<tr>
<td>PRICE: $9,200</td>
<td>KMS:</td>
<td></td>
</tr>
<tr>
<td>BODY_TYPE: WAGON</td>
<td>PRICE: $5,990</td>
<td></td>
</tr>
<tr>
<td>COLOUR:</td>
<td>STATE: SA</td>
<td></td>
</tr>
<tr>
<td>LOCATION: NEWCASTLE AND CENTRAL</td>
<td>DEALER: Auto Credit Co</td>
<td></td>
</tr>
<tr>
<td><strong>VEHICLE</strong></td>
<td><strong>CAR</strong></td>
<td></td>
</tr>
<tr>
<td>YEAR: 1995</td>
<td>MAKE_MODEL: Subaru Liberty GX</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION: HOLDEN BARINA SWING</td>
<td>YR: 1993</td>
<td></td>
</tr>
<tr>
<td>PRICE: $6,995</td>
<td>KMS:</td>
<td></td>
</tr>
<tr>
<td>BODY_TYPE:</td>
<td>PRICE: $9,990</td>
<td></td>
</tr>
<tr>
<td>COLOUR:</td>
<td>STATE: VIC</td>
<td></td>
</tr>
<tr>
<td>LOCATION: NEWCASTLE AND CENTRAL</td>
<td>DEALER: Eastern Vehicle</td>
<td></td>
</tr>
<tr>
<td><strong>VEHICLE</strong></td>
<td><strong>CAR</strong></td>
<td></td>
</tr>
<tr>
<td>YEAR: 1994</td>
<td>MAKE_MODEL: Ford Falcon XH</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION: SUBARU SPORTSWAGON</td>
<td>YR: 1997</td>
<td></td>
</tr>
<tr>
<td>PRICE: $7,990</td>
<td>KMS:</td>
<td></td>
</tr>
<tr>
<td>BODY_TYPE:</td>
<td>PRICE: $9,999</td>
<td></td>
</tr>
<tr>
<td>COLOUR:</td>
<td>STATE: VIC</td>
<td></td>
</tr>
<tr>
<td>LOCATION:</td>
<td>DEALER: Lucky Choice Ca</td>
<td></td>
</tr>
<tr>
<td><strong>VEHICLE</strong></td>
<td><strong>CAR</strong></td>
<td></td>
</tr>
<tr>
<td>YEAR: 1996</td>
<td>MAKE_MODEL: Mitsubishi Magna</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION: MITSUBISHI LANCER GLI CE</td>
<td>YR: 1991</td>
<td></td>
</tr>
<tr>
<td>PRICE: $9,990</td>
<td>KMS: 176,330</td>
<td></td>
</tr>
<tr>
<td>BODY_TYPE: COUPE</td>
<td>PRICE: $9,000</td>
<td></td>
</tr>
<tr>
<td>COLOUR: METALLIC BLUE</td>
<td>STATE: NSW</td>
<td></td>
</tr>
<tr>
<td>LOCATION:</td>
<td>DEALER: Laurieton Motor</td>
<td></td>
</tr>
<tr>
<td>VEHICLE</td>
<td>YEAR</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>1992</td>
<td>MITSUBISHI MAGNA</td>
</tr>
<tr>
<td></td>
<td>1996</td>
<td>FORD FALCON</td>
</tr>
<tr>
<td></td>
<td>1995</td>
<td>HOLDEN BARINA SWING</td>
</tr>
<tr>
<td></td>
<td>1994</td>
<td>SUBARU SPORTSWAGON</td>
</tr>
<tr>
<td></td>
<td>1996</td>
<td>MITSUBISHI LANCER GLI CE</td>
</tr>
<tr>
<td></td>
<td>1994</td>
<td>MITSUBISHI MAGNA TS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VEHICLE</th>
<th>YEAR</th>
<th>DESCRIPTION</th>
<th>PRICE</th>
<th>BODY_TYPE</th>
<th>COLOUR</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1992</td>
<td>MITSUBISHI MAGNA</td>
<td>$9,990</td>
<td>SEDAN</td>
<td>WHITE/GREY</td>
<td>HAWTHORN, VIC</td>
</tr>
<tr>
<td></td>
<td>1996</td>
<td>FORD FALCON</td>
<td>$9,200</td>
<td>WAGON</td>
<td></td>
<td>NEWCASTLE AND CENTRAL</td>
</tr>
<tr>
<td></td>
<td>1995</td>
<td>HOLDEN BARINA SWING</td>
<td>$6,995</td>
<td></td>
<td></td>
<td>NEWCASTLE AND CENTRAL</td>
</tr>
<tr>
<td></td>
<td>1994</td>
<td>SUBARU SPORTSWAGON</td>
<td>$7,990</td>
<td></td>
<td></td>
<td>SYDNEY</td>
</tr>
<tr>
<td></td>
<td>1996</td>
<td>MITSUBISHI LANCER GLI CE</td>
<td>$9,990</td>
<td>COUPE</td>
<td>METALLIC BLUE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1994</td>
<td>MITSUBISHI MAGNA TS</td>
<td>$7,995</td>
<td>SEDAN</td>
<td>MAROON</td>
<td>MT. GAMBIER, SA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AutoWeb</th>
<th>AutoWeb</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**VEHICLE**

<table>
<thead>
<tr>
<th>YEAR</th>
<th>DESCRIPTION</th>
<th>PRICE</th>
<th>BODY_TYPE</th>
<th>COLOUR</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td>MITSUBISHI MAGNA</td>
<td>$9,990</td>
<td>SEDAN</td>
<td>WHITE/GREY</td>
<td>HAWTHORN, VIC</td>
</tr>
<tr>
<td>1996</td>
<td>FORD FALCON</td>
<td>$9,200</td>
<td>WAGON</td>
<td></td>
<td>NEWCASTLE AND CENTRAL</td>
</tr>
<tr>
<td>1995</td>
<td>HOLDEN BARINA SWING</td>
<td>$6,995</td>
<td></td>
<td></td>
<td>NEWCASTLE AND CENTRAL</td>
</tr>
<tr>
<td>1996</td>
<td>SUBARU SPORTSWAGON</td>
<td>$7,990</td>
<td>COUPE</td>
<td>METALLIC BLUE</td>
<td>SYDNEY</td>
</tr>
<tr>
<td>1994</td>
<td>MITSUBISHI MAGNA TS</td>
<td>$7,995</td>
<td>SEDAN</td>
<td>MAROON</td>
<td>MT. GAMBIER, SA</td>
</tr>
</tbody>
</table>

**CAR**

| Unknown1 | 1986 MAZDA RX7 FC1031 Limited, 2 door COUPE, white, 4 sp Automatic, 170000km, Petrol Rotary 1.3 2cyl (1308cc). Reg. No UDK-569 |
| Unknown2 | $6,500 |
| Unknown3 | Only $6500 ONO. price is negotiable. the 5th picture is what the car COULD look like. |
| Unknown4 | Fulham Gardens, Adelaide, SA |

| Unknown1 | 1993 FORD LASER KH S, 5 door HATCHBACK, green, 5 sp Manual, 245000km, Petrol Multi-point injected 1.8 4cyl (1840cc). Reg. No NBS 437 |
| Unknown2 | $6,000 |
| Unknown3 | laser 93KH sports,1.8 EFI, air cond,p-steer,t-windows, $6000 |
| Unknown4 | Wodonga, VIC |

| Unknown1 | 1997 PROTON SATRIA GLi, 3 door HATCHBACK, GREEN, 5 sp Manual, 112000km, Petrol Multi-point injected 1.5 4cyl (1468cc). Reg. No UNH-993 |
| Unknown2 | $5,950 |
| Unknown3 | Good clean economical car - priced to sell |
| Unknown4 | Castle Hill, NSW |

| Unknown1 | 1984 MERCEDES 230E W123, 4 door SEDAN, Champagne (gold), 4 sp Automatic, 220000km, Petrol Multi-point injected 2.3 4cyl (2299cc). Reg. No OIL363 |
| Unknown2 | $6,500 |
| Unknown3 | |
| Unknown4 | Brighton, VIC |

| Unknown1 | 1992 FORD FALCON EB II GLi, 4 door WAGON, red, 4 sp Automatic, 200000km, Petrol Multi-point injected 4.0 6cyl (3984cc). Reg. No 415GTC |
| Unknown2 | $5,100 |
| Unknown3 | 6 seater |
| Unknown4 | Warner Brisbane, QLD |
|---------|-----------|--------------------------------|--------------|-----------------|------------------|-------------------------|
| VEHICLE | YEAR: 1996 | DESCRIPTION: MITSUBISHI LANCER GLI CE | PRICE: $9,990 | BODY_TYPE: COUPE | COLOUR: METALLIC BLUE | LOCATION: |


<table>
<thead>
<tr>
<th>AutoWeb</th>
<th>Drive</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VEHICLE</td>
<td>YEAR</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>1992</td>
<td>MITSUBISHI MAGNA</td>
</tr>
<tr>
<td></td>
<td>1996</td>
<td>FORD FALCON</td>
</tr>
<tr>
<td></td>
<td>1995</td>
<td>HOLDEN BARINA SWING</td>
</tr>
<tr>
<td></td>
<td>1994</td>
<td>SUBARU SPORTSWAGON</td>
</tr>
<tr>
<td></td>
<td>1996</td>
<td>MITSUBISHI LANCER GLI CE</td>
</tr>
<tr>
<td></td>
<td>1994</td>
<td>MITSUBISHI MAGNA TS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VEHICLE</th>
<th>BIDS</th>
<th>CAR</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>Ford Fairlane 1991 NC</td>
<td>AU $6,000.00</td>
<td>4d 02h 33m</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>MUST BE SEEN. SERIES II VR SS V8 5 SPEED</td>
<td>AU $12,100.00</td>
<td>17h 30m</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>BMW E30 Series John Player Special 323i Coupe</td>
<td>AU $10,950.00</td>
<td>9d 16h 51m</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>FORD Laser KF TX3 1.8 ltr - 1990</td>
<td>AU $2,560.00</td>
<td>2d 19h 02m</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Mitsubishi L300 van</td>
<td>AU $1,000.00</td>
<td>2d 01h 14m</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>69 Camaro 396 RS/SS Matching # car</td>
<td>AU $40,000.00</td>
<td>5d 00h 29m</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>fuel saver economy increases lowers emissions</td>
<td>AU $35.00</td>
<td>6d 01h 56m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AutoWeb vs eBay</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Carsales vs Autotrader**

<table>
<thead>
<tr>
<th>MAKE_MODEL</th>
<th>YR</th>
<th>KMS</th>
<th>PRICE</th>
<th>STATE</th>
<th>DEALER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mazda 323 Astina</td>
<td>1992</td>
<td>140,000</td>
<td>$9,999</td>
<td>QLD</td>
<td>Ferrierr Qualit</td>
</tr>
<tr>
<td>Saab 9000</td>
<td>1987</td>
<td>74,500</td>
<td>$5,990</td>
<td>SA</td>
<td>Auto Credit Co</td>
</tr>
<tr>
<td>Subaru Liberty GX</td>
<td>1993</td>
<td>222,500</td>
<td>$9,990</td>
<td>VIC</td>
<td>Eastern Vehicle</td>
</tr>
<tr>
<td>Ford Falcon XH</td>
<td>1997</td>
<td>243,000</td>
<td>$9,990</td>
<td>VIC</td>
<td>Lucky Choice Ca</td>
</tr>
<tr>
<td>Ford Fairlane</td>
<td>1991</td>
<td>176,330</td>
<td>$9,000</td>
<td>NSW</td>
<td></td>
</tr>
<tr>
<td>Mitsubishi Magna</td>
<td>1994</td>
<td>142,000</td>
<td>$8,990</td>
<td>NSW</td>
<td>Laurieton Motor</td>
</tr>
</tbody>
</table>

**Carsales vs Autotrader Confidence (%)**

<table>
<thead>
<tr>
<th>MAKE_MODEL</th>
<th>YR</th>
<th>KMS</th>
<th>PRICE</th>
<th>STATE</th>
<th>DEALER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mazda 323 Astina</td>
<td>1992</td>
<td>140,000</td>
<td>$9,999</td>
<td>QLD</td>
<td>Ferrierr Qualit</td>
</tr>
<tr>
<td>Saab 9000</td>
<td>1987</td>
<td>74,500</td>
<td>$5,990</td>
<td>SA</td>
<td>Auto Credit Co</td>
</tr>
<tr>
<td>Subaru Liberty GX</td>
<td>1993</td>
<td>222,500</td>
<td>$9,990</td>
<td>VIC</td>
<td>Eastern Vehicle</td>
</tr>
<tr>
<td>Ford Falcon XH</td>
<td>1997</td>
<td>243,000</td>
<td>$9,990</td>
<td>VIC</td>
<td>Lucky Choice Ca</td>
</tr>
<tr>
<td>Ford Fairlane</td>
<td>1991</td>
<td>176,330</td>
<td>$9,000</td>
<td>NSW</td>
<td></td>
</tr>
<tr>
<td>Mitsubishi Magna</td>
<td>1994</td>
<td>142,000</td>
<td>$8,990</td>
<td>NSW</td>
<td>Laurieton Motor</td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1: 1986 MAZDA RX7 FC1031 Limited, 2 door COUPE, white, 4 sp Automatic, 170000km, Petrol Rotary 1.3 2cyl (1308cc). Reg. No UDK-569</td>
<td>Unknown2: $6,500</td>
<td>Unknown3: Only $6500 ONO. price is negotiable. the 5th picture is what the car COULD look like.</td>
<td>Unknown4: Fulham Gardens, Adelaide, SA</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1: 1993 FORD LASER KH S, 5 door HATCHBACK, green, 5 sp Manual, 245000km, Petrol Multi-point injected 1.8 4cyl (1840cc). Reg. No NBS 437</td>
<td>Unknown2: $6,000</td>
<td>Unknown3: lasar 93KH sports, 1.8 EFI, air cond, p-steer, t-windows, $6000</td>
<td>Unknown4: Woodonga, VIC</td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1: 1997 PROTON SATRIA GLi, 3 door HATCHBACK, GREEN, 5 sp Manual, 112000km, Petrol Multi-point injected 1.5 4cyl (1468cc). Reg. No UNH-993</td>
<td>Unknown2: $5,950</td>
<td>Unknown3: Good clean economical car - priced to sell</td>
<td>Unknown4: Castle Hill, NSW</td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1: 1984 MERCEDES 230E W123, 4 door SEDAN, Champagne (gold), 4 sp Automatic, 220000km, Petrol Multi-point injected 2.3 4cyl (2299cc). Reg. No OIL363</td>
<td>Unknown2: $6,500</td>
<td>Unknown3:</td>
<td>Unknown4: Brighton, VIC</td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1: 1992 FORD FALCON EB II GLi, 4 door WAGON, red, 4 sp Automatic, 200000km, Petrol Multi-point injected 4.0 6cyl (3984cc). Reg. No 415GTC</td>
<td>Unknown2: $5,100</td>
<td>Unknown3: 6 seater</td>
<td>Unknown4: Warner Brisbane, QLD</td>
<td></td>
</tr>
</tbody>
</table>

| VEHICLE | YEAR: 1996 | DESCRIPTION: MITSUBISHI LANCER GLi CE | PRICE: $9,990 | BODY_TYPE: COUPE | COLOUR: METALLIC BLUE | LOCATION: |
| CAR | Unknown1: 1986 MAZDA RX7 FC1031 Limited, 2 door COUPE, white, 4 sp Automatic, 170000km, Petrol Rotary 1.3 2cyl (1308cc). Reg. No UDK-569 | Unknown1: 1986 MAZDA RX7 FC1031 Limited, 2 door COUPE, white, 4 sp Automatic, 170000km, Petrol Rotary 1.3 2cyl (1308cc). Reg. No UDK-569 | Unknown2: $6,500 | Unknown2: $6,500 |
| Unknown3: Only $6500 ONO. price is negotiable. the 5th picture is what the car COULD look like. | Unknown3: Only $6500 ONO. price is negotiable. the 5th picture is what the car COULD look like. | Unknown4: Fulham Gardens, Adelaide, SA | Unknown4: Fulham Gardens, Adelaide, SA |
| CAR | Unknown1: 1993 FORD LASER KH S, 5 door HATCHBACK, green, 5 sp Manual, 245000km, Petrol Multi-point injected 1.8 4cyl (1840cc). Reg. No NBS 437 | Unknown1: 1993 FORD LASER KH S, 5 door HATCHBACK, green, 5 sp Manual, 245000km, Petrol Multi-point injected 1.8 4cyl (1840cc). Reg. No NBS 437 | Unknown2: $6,000 | Unknown2: $6,000 |
| Unknown3: laser 93KH sports,1.8 EFI, air cond,p- steer,t-windows, $6000 | Unknown3: laser 93KH sports,1.8 EFI, air cond,p-steer,t-windows, $6000 | Unknown4: Wodonga, VIC | Unknown4: Wodonga, VIC |
| CAR | Unknown1: 1997 PROTON SATRIA GLi, 3 door HATCHBACK, GREEN, 5 sp Manual, 112000km, Petrol Multi-point injected 1.5 4cyl (1468cc). Reg. No UNH-993 | Unknown1: 1997 PROTON SATRIA GLi, 3 door HATCHBACK, GREEN, 5 sp Manual, 112000km, Petrol Multi-point injected 1.5 4cyl (1468cc). Reg. No UNH-993 | Unknown2: $5,950 | Unknown2: $5,950 |
| Unknown3: Good clean economical car - priced to sell | Unknown3: Good clean economical car - priced to sell | Unknown4: Castle Hill, NSW | Unknown4: Castle Hill, NSW |
| CAR | Unknown1: 1984 MERCEDES 230E W123, 4 door SEDAN, Champagne (gold), 4 sp Automatic, 220000km, Petrol Multi-point injected 2.3 4cyl (2299cc). Reg. No OIL363 | Unknown1: 1984 MERCEDES 230E W123, 4 door SEDAN, Champagne (gold), 4 sp Automatic, 220000km, Petrol Multi-point injected 2.3 4cyl (2299cc). Reg. No OIL363 | Unknown2: $6,500 | Unknown2: $6,500 |
| Unknown3: $6,500 | Unknown3: $6,500 | Unknown4: Brighton, VIC | Unknown4: Brighton, VIC |
| CAR | Unknown1: 1992 FORD FALCON EB II GLi, 4 door WAGON, red, 4 sp Automatic, 200000km, Petrol Multi-point injected 4.0 6cyl (3984cc). Reg. No 415GTC | Unknown1: 1992 FORD FALCON EB II GLi, 4 door WAGON, red, 4 sp Automatic, 200000km, Petrol Multi-point injected 4.0 6cyl (3984cc). Reg. No 415GTC | Unknown2: $5,100 | Unknown2: $5,100 |
| Carsales | Carsales | Confidence (%) | Carsales |

<p>| Carsales | Carsales | Confidence (%) | Carsales |</p>
<table>
<thead>
<tr>
<th>Carsales</th>
<th>Drive</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CAR</strong></td>
<td>Unknown1: 1986 MAZDA RX7 FC1031 Limited, 2 door COUPE, white, 4 sp Automatic, 170000km, Petrol Rotary 1.3 2cyl (1308cc). Reg. No UDK-569</td>
<td></td>
</tr>
<tr>
<td>Unknown2: $6,500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown3: Only $6500 ONO. price is negotiable. the 5th picture is what the car COULD look like.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown4: Fulham Gardens, Adelaide, SA</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CAR</strong></td>
<td>Unknown1: 1993 FORD LASER KH S, 5 door HATCHBACK, green, 5 sp Manual, 245000km, Petrol Multi-point injected 1.8 4cyl (1840cc). Reg. No NBS 437</td>
<td></td>
</tr>
<tr>
<td>Unknown2: $6,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown3: laser 93KH sports, 1.8 EFI, air cond, p-steer, t-windows, $6000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown4: Wodonga, VIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CAR</strong></td>
<td>Unknown1: 1997 PROTON SATRIA GLi, 3 door HATCHBACK, GREEN, 5 sp Manual, 112000km, Petrol Multi-point injected 1.5 4cyl (1468cc). Reg. No UNH-993</td>
<td></td>
</tr>
<tr>
<td>Unknown2: $5,950</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown3: Good clean economical car - priced to sell</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown4: Castle Hill, NSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CAR</strong></td>
<td>Unknown1: 1984 MERCEDES 230E W123, 4 door SEDAN, Champagne (gold), 4 sp Automatic, 220000km, Petrol Multi-point injected 2.3 4cyl (2299cc). Reg. No OIL363</td>
<td></td>
</tr>
<tr>
<td>Unknown2: $6,500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown3:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown4: Brighton, VIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CAR</strong></td>
<td>Unknown1: 1992 FORD FALCON EB II GLi, 4 door WAGON, red, 4 sp Automatic, 200000km, Petrol Multi-point injected 4.0 6cyl (3984cc). Reg. No 415GTC</td>
<td></td>
</tr>
<tr>
<td>Unknown2: $5,100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown3: 6 seater</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown4: Warner Brisbane, QLD</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CAR</strong></td>
<td>Unknown1: MERCEDES-BENZ 380</td>
<td></td>
</tr>
<tr>
<td>Unknown2: $9500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>year:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>kms:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trans:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>body:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
<td></td>
</tr>
<tr>
<td>features:</td>
<td>Anti lock braking, Alarm system, Cruise control, Radio cassette...</td>
<td></td>
</tr>
<tr>
<td>description:</td>
<td>MERCEDES 380SEL, Diamond Blue, blue lther, recond motor, Alpine stereo, alarm, ABS, sunrf, cruise, reg 1/04 YDQ756. $9500 ono 0439 311 767.</td>
<td></td>
</tr>
<tr>
<td>location:</td>
<td>The Sydney Morning Herald</td>
<td></td>
</tr>
<tr>
<td><strong>CAR</strong></td>
<td>Unknown1: HOLDEN COMMODORE</td>
<td></td>
</tr>
<tr>
<td>Unknown2: $7250</td>
<td></td>
<td></td>
</tr>
<tr>
<td>year:</td>
<td>1992</td>
<td></td>
</tr>
<tr>
<td>kms:</td>
<td>175000</td>
<td></td>
</tr>
<tr>
<td>trans:</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>body:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
<td></td>
</tr>
<tr>
<td>features:</td>
<td>Power steering, Air conditioning...</td>
<td></td>
</tr>
<tr>
<td>description:</td>
<td>COMMODORE VP, '93 Series 11, auto., air, steer, $5200 ono. YNR-126. PH. 4959 1992</td>
<td></td>
</tr>
<tr>
<td>location:</td>
<td>The Newcastle Herald and Post</td>
<td></td>
</tr>
<tr>
<td><strong>CAR</strong></td>
<td>Unknown1: HOLDEN COMMODORE</td>
<td></td>
</tr>
<tr>
<td>Unknown2: $6500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>year:</td>
<td>1991</td>
<td></td>
</tr>
<tr>
<td>kms:</td>
<td>55000</td>
<td></td>
</tr>
<tr>
<td>trans:</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>body:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
<td></td>
</tr>
<tr>
<td>features:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description:</td>
<td>TOYOTA CAMRY ’91 EXECUTIVE AUTO ONE OWNER 53,000 k's $6500. RWK-150. PH: 4946 5548</td>
<td></td>
</tr>
<tr>
<td>location:</td>
<td>The Newcastle Herald and Post</td>
<td></td>
</tr>
<tr>
<td><strong>CAR</strong></td>
<td>Unknown1: TOYOTA CAMRY</td>
<td></td>
</tr>
<tr>
<td>Unknown2: $6500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>year:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kms:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trans:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>body:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>features:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>location:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Carsales vs eBay

<table>
<thead>
<tr>
<th>CAR</th>
<th>Unknown1: 1986 MAZDA RX7 FC1031 Limited, 2 door COUPE, white, 4 sp Automatic, 170000km, Petrol Rotary 1.3 2cyl (1308cc). Reg. No UDQ-569</th>
<th>VEHICLE</th>
<th>BIDS: 0</th>
<th>CAR: Ford Fairlane 1991 NC</th>
<th>PRICE: AU $6,000.00</th>
<th>TIME_LEFT: 4d 02h 33m</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unknown2: $6,500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown3: Only $6500 ONO. price is negotiable. the 5th picture is what the car COULD look like.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown4: Fulham Gardens, Adelaide, SA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1: 1993 FORD LASER KH S, 5 door HATCHBACK, green, 5 sp Manual, 245000km, Petrol Multi-point injected 1.8 4cyl (1840cc). Reg. No NBS 437</td>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: MITSUBISHI MAGNA TJ SPORTS 2001 MODEL</td>
<td>PRICE: AU $25,000.00</td>
<td>TIME_LEFT: 6d 13h 30m</td>
</tr>
<tr>
<td></td>
<td>Unknown2: $6,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown3: laser 93KH sports, 1.8 EFI, air cond, t- steer, t-windows, $6000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown4: Wodonga, VIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1: 1997 PROTON SATRIA GLi, 3 door HATCHBACK, GREEN, 5 sp Manual, 112000km, Petrol Multi-point injected 1.5 4cyl (1468cc). Reg. No UNH-993</td>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: 1985 Holden VK Commodore Executive Unreg</td>
<td>PRICE: AU $500.00</td>
<td>TIME_LEFT: 6d 14h 12m</td>
</tr>
<tr>
<td></td>
<td>Unknown2: $5,950</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown3: Good clean economical car - priced to sell</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown4: Castle Hill, NSW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1: 1984 MERCEDES 230E W123, 4 door SEDAN, Champagne (gold), 4 sp Automatic, 220000km, Petrol Multi-point injected 2.3 4cyl (2299cc). Reg. No OIL363</td>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: BMW E30 Series John Player Special 323i Coupe</td>
<td>PRICE: AU $10,950.00</td>
<td>TIME_LEFT: 9d 16h 51m</td>
</tr>
<tr>
<td></td>
<td>Unknown2: $6,500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown3:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown4: Brighton, VIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1: 1992 FORD FALCON EB II GLi, 4 door WAGON, red, 4 sp Automatic, 200000km, Petrol Multi-point injected 4.0 6cyl (3984cc). Reg. No 415GTC</td>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: MUST BE SEEN. SERIES II VR SS V8 5 SPEED</td>
<td>PRICE: AU $12,100.00</td>
<td>TIME_LEFT: 17h 30m</td>
</tr>
<tr>
<td></td>
<td>Unknown2: $5,100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown3: 6 seater</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown4: Warner Brisbane, QLD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Carsales</th>
<th>eBay</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Drive vs Autotrader

<table>
<thead>
<tr>
<th>Make/Model</th>
<th>Year</th>
<th>KMS</th>
<th>Price</th>
<th>State</th>
<th>Dealer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mazda 323 Astina</td>
<td>1992</td>
<td>140,000</td>
<td>$9,999</td>
<td>QLD</td>
<td>Ferrier Quality</td>
</tr>
<tr>
<td>Saab 9000</td>
<td>1987</td>
<td></td>
<td>$5,990</td>
<td>SA</td>
<td>Auto Credit Co</td>
</tr>
<tr>
<td>Subaru Liberty GX</td>
<td>1993</td>
<td></td>
<td>$9,990</td>
<td>VIC</td>
<td>Eastern Vehicle</td>
</tr>
<tr>
<td>Ford Falcon XH</td>
<td>1997</td>
<td></td>
<td>$9,999</td>
<td>VIC</td>
<td>Lucky Choice</td>
</tr>
<tr>
<td>Ford Fairlane</td>
<td>1991</td>
<td></td>
<td>$9,000</td>
<td>NSW</td>
<td></td>
</tr>
<tr>
<td>Mitsubishi Magna</td>
<td>1994</td>
<td>142,000</td>
<td>$8,990</td>
<td>NSW</td>
<td>Laurieton Motor</td>
</tr>
</tbody>
</table>
### Drive vs AutoWeb

**CAR**

**Unknown1:** MERCEDES-BENZ 380  
**Unknown2:** $9500  
**year:** 0  
**kms:**  
**trans:**  
**body:**  
**colour:**  
**ad_date:** 03/05/2003  
**features:** Anti lock braking, Alarm system, Cruise control, Radio cassette,  
**description:** MERCEDES 380SEL, DIamond Blue, blue leather, recond motor, Alpine stereo, alarm, ABS, sunroof, cruise, reg 1/04 YDQ756. $9500 ono 0439 311 767.  
**location:** The Sydney Morning Herald

**CAR**

**Unknown1:** HOLDEN COMMODORE  
**Unknown2:** $5200  
**year:** 1993  
**kms:**  
**trans:** A  
**body:**  
**colour:**  
**ad_date:** 03/05/2003  
**features:** Power steering, Air conditioning...  
**description:** COMMODORE VP, '93 Series 11, auto., air, steer, $5200 ono. YNR-126, PH: 4959 1992  
**location:** The Newcastle Herald and Post

**CAR**

**Unknown1:** HOLDEN COMMODORE  
**Unknown2:** $7250  
**year:** 1992  
**kms:** 175000  
**trans:**  
**body:**  
**colour:**  
**ad_date:** 03/05/2003  
**features:**  
**description:** COMMODORE VP 1992 Series II, TSU-387, regd Nov '03, new paint, tyres, very good cond, 175,000 ks, $7250 ono. 4965 9112.  
**location:** The Newcastle Herald and Post

**CAR**

**Unknown1:** TOYOTA CAMRY  
**Unknown2:** $6500  
**year:** 1991  
**kms:** 53000  
**trans:** A  
**body:**  
**colour:**  
**ad_date:** 03/05/2003  
**features:**  
**description:** TOYOTA CAMRY '91 EXECUTIVE AUTO ONE OWNER 53,000 k's $6500. RWK-150. PH: 4946 5548  
**location:** The Newcastle Herald and Post

---

**VEHICLE**

**YEAR:** 1992  
**DESCRIPTION:** MITSUBISHI MAGNA  
**PRICE:** $9,990  
**BODY_TYPE:** SEDAN  
**COLOUR:** WHITE/GREY  
**LOCATION:** HAWTHORN, VIC

**VEHICLE**

**YEAR:** 1996  
**DESCRIPTION:** FORD FALCON  
**PRICE:** $9,200  
**BODY_TYPE:** WAGON  
**COLOUR:**  
**LOCATION:** NEWCASTLE AND CENTRAL

**VEHICLE**

**YEAR:** 1995  
**DESCRIPTION:** HOLDEN BARINA SWING  
**PRICE:** $6,995  
**BODY_TYPE:**  
**COLOUR:**  
**LOCATION:** NEWCASTLE AND CENTRAL

**VEHICLE**

**YEAR:** 1994  
**DESCRIPTION:** SUBARU SPORTSWAGON  
**PRICE:** $7,990  
**BODY_TYPE:**  
**COLOUR:**  
**LOCATION:** SYDNEY

**VEHICLE**

**YEAR:** 1996  
**DESCRIPTION:** MITSUBISHI LANCER GLI CE  
**PRICE:** $9,990  
**BODY_TYPE:** COUPE  
**COLOUR:** METALLIC BLUE  
**LOCATION:**

**VEHICLE**

**YEAR:** 1994  
**DESCRIPTION:** MITSUBISHI MAGNA TS  
**PRICE:** $7,995  
**BODY_TYPE:** SEDAN  
**COLOUR:** MAROON  
**LOCATION:** MT. GAMBIER, SA

---

<table>
<thead>
<tr>
<th>Drive</th>
<th>AutoWeb</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1:</td>
<td>MERCEDES-BENZ 380</td>
</tr>
<tr>
<td></td>
<td>Unknown2:</td>
<td>$9500</td>
</tr>
<tr>
<td></td>
<td>year:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>kms:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>trans:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>body:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td></td>
<td>features:</td>
<td>Anti lock braking, Alarm system, Cruise control, Radio cassette,</td>
</tr>
<tr>
<td></td>
<td>location:</td>
<td>The Sydney Morning Herald</td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1:</td>
<td>HOLDEN COMMODORE</td>
</tr>
<tr>
<td></td>
<td>Unknown2:</td>
<td>$5200</td>
</tr>
<tr>
<td></td>
<td>year:</td>
<td>1993</td>
</tr>
<tr>
<td></td>
<td>kms:</td>
<td>175000</td>
</tr>
<tr>
<td></td>
<td>trans:</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>body:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td></td>
<td>features:</td>
<td>Power steering, Air conditioning...</td>
</tr>
<tr>
<td></td>
<td>description:</td>
<td>COMMODORE VP, ’93 Series 11, auto., air, steer, $5200 ono. YNR-126, PH: 4959 1992</td>
</tr>
<tr>
<td></td>
<td>location:</td>
<td>The Newcastle Herald and Post</td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1:</td>
<td>HOLDEN COMMODORE</td>
</tr>
<tr>
<td></td>
<td>Unknown2:</td>
<td>$7250</td>
</tr>
<tr>
<td></td>
<td>year:</td>
<td>1992</td>
</tr>
<tr>
<td></td>
<td>kms:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>trans:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>body:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td></td>
<td>features:</td>
<td>COMMODEROE VP 1992 Series II, TSU-387, regd Nov '03, new paint, tyres, very good cond, 175,000 ks, $7250 ono. 4965 9112.</td>
</tr>
<tr>
<td></td>
<td>description:</td>
<td>COMMODORE VP, ’93 Series 11, auto., air, steer, $5200 ono. YNR-126, PH: 4959 1992</td>
</tr>
<tr>
<td></td>
<td>location:</td>
<td>The Newcastle Herald and Post</td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1:</td>
<td>TOYOTA CAMRY</td>
</tr>
<tr>
<td></td>
<td>Unknown2:</td>
<td>$6500</td>
</tr>
<tr>
<td></td>
<td>year:</td>
<td>1991</td>
</tr>
<tr>
<td></td>
<td>kms:</td>
<td>53000</td>
</tr>
<tr>
<td></td>
<td>trans:</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>body:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td></td>
<td>features:</td>
<td>TOYOTA CAMRY ’91 EXECUTIVE AUTO ONE OWNER 53,000 k’s $6500. RWK-150. PH: 4946 5548</td>
</tr>
<tr>
<td></td>
<td>location:</td>
<td>The Newcastle Herald and Post</td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1:</td>
<td>1986 MAZDA RX7 FC1031 Limited, 2 door COUPE, white, 4 sp Automatic, 170000km , Petrol Rotary 1.3 2cyl (1308cc). Reg. No UDK-569</td>
</tr>
<tr>
<td></td>
<td>Unknown2:</td>
<td>$6,500</td>
</tr>
<tr>
<td></td>
<td>Unknown3:</td>
<td>Only $6500 ONO. price is negotiable. the 5th picture is what the car COULD look like.</td>
</tr>
<tr>
<td></td>
<td>Unknown4:</td>
<td>Fulham Gardens, Adelaide, SA</td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1:</td>
<td>1993 FORD LASER KH S, 5 door HATCHBACK, green, 5 sp Manual, 245000km , Petrol Multi-point injected 1.8 4cyl (1840cc). Reg. No NBS 437</td>
</tr>
<tr>
<td></td>
<td>Unknown2:</td>
<td>$6,000</td>
</tr>
<tr>
<td></td>
<td>Unknown3:</td>
<td>laser 93KH sports,1.8 EFI, air cond,p-steer,t-windows, $6000</td>
</tr>
<tr>
<td></td>
<td>Unknown4:</td>
<td>Wodonga, VIC</td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1:</td>
<td>1997 PROTON SATRIA GLi, 3 door HATCHBACK, GREEN, 5 sp Manual, 112000km , Petrol Multi-point injected 1.5 4cyl (1468cc). Reg. No UNH-993</td>
</tr>
<tr>
<td></td>
<td>Unknown2:</td>
<td>$5,950</td>
</tr>
<tr>
<td></td>
<td>Unknown3:</td>
<td>Good clean economical car - priced to sell</td>
</tr>
<tr>
<td></td>
<td>Unknown4:</td>
<td>Castle Hill, NSW</td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1:</td>
<td>1984 MERCEDES 230E W123 , 4 door SEDAN, Champagne (gold), 4 sp Automatic, 220000km , Petrol Multi-point injected 2.3 4cyl (2299cc). Reg. No OIL363</td>
</tr>
<tr>
<td></td>
<td>Unknown2:</td>
<td>$6,500</td>
</tr>
<tr>
<td></td>
<td>Unknown3:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown4:</td>
<td>Brighton, VIC</td>
</tr>
<tr>
<td>CAR</td>
<td>Unknown1:</td>
<td>1992 FORD FALCON EB II GLi, 4 door WAGON, red, 4 sp Automatic, 200000km , Petrol Multi-point injected 4.0 6cyl (3984cc). Reg. No 415GTC</td>
</tr>
<tr>
<td></td>
<td>Unknown2:</td>
<td>$5,100</td>
</tr>
<tr>
<td></td>
<td>Unknown3:</td>
<td>6 seater</td>
</tr>
<tr>
<td></td>
<td>Unknown4:</td>
<td>Warner Brisbane, QLD</td>
</tr>
</tbody>
</table>
CAR
Unknown1: MERCEDES-BENZ 380
Unknown2: $9500
year: 0
kms: 
trans: 
body: 
colour: 
ad_date: 03/05/2003
features: Anti lock braking, Alarm system, Cruise control, Radio cassette...
description: MERCEDES 380SEL, Diamond Blue, blue lther, recond motor, Alpine stereo, alarm, ABS, sunrf, cruise, reg 1/04 YDQ756. $9500 ono 0439 311 767. location: The Sydney Morning Herald

CAR
Unknown1: HOLDEN COMMODORE
Unknown2: $5200
year: 1993
kms: 
trans: A
body: 
colour: 
ad_date: 03/05/2003
features: Power steering, Air conditioning...

CAR
Unknown1: HOLDEN COMMODORE
Unknown2: $7250
year: 1992
kms: 175000
trans: 
body: 
colour: 
ad_date: 03/05/2003
features: 
description: COMMODORE VP 1992 Series II, TSU-387, regd Nov '03, new paint, tyres, very good cond, 175,000 ks, $7250 ono. 4965 9112. location: The Newcastle Herald and Post

CAR
Unknown1: TOYOTA CAMRY
Unknown2: $6500
year: 1991
kms: 53000
trans: A
body: 
colour: 
ad_date: 03/05/2003
features: 
description: TOYOTA CAMRY '91 EXECUTIVE AUTO ONE OWNER 53,000 k's $6500. RWK-150. PH: 4946 5548 location: The Newcastle Herald and Post

<table>
<thead>
<tr>
<th>Drive</th>
<th>Drive</th>
<th>Confidence (%)</th>
</tr>
</thead>
</table>
## Drive vs eBay

### CAR

<table>
<thead>
<tr>
<th>Unknown1:</th>
<th>MERCEDES-BENZ 380</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown2:</td>
<td>$9500</td>
</tr>
<tr>
<td>year:</td>
<td>0</td>
</tr>
<tr>
<td>kms:</td>
<td></td>
</tr>
<tr>
<td>trans:</td>
<td></td>
</tr>
<tr>
<td>body:</td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td>features:</td>
<td>Anti lock braking, Alarm system, Cruise control, Radio cassette,..</td>
</tr>
<tr>
<td>description:</td>
<td>MERCEDES 380SEL, Diamond Blue, blue lther, recd motor, Alpine stereo, alarm, ABS, sunrf, cruise, reg 1/04 YDQ756. $9500 ono 0439 311 767.</td>
</tr>
<tr>
<td>location:</td>
<td>The Sydney Morning Herald</td>
</tr>
</tbody>
</table>

### CAR

<table>
<thead>
<tr>
<th>Unknown1:</th>
<th>HOLDEN COMMODORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown2:</td>
<td>$5200</td>
</tr>
<tr>
<td>year:</td>
<td>1993</td>
</tr>
<tr>
<td>kms:</td>
<td></td>
</tr>
<tr>
<td>trans:</td>
<td>A</td>
</tr>
<tr>
<td>body:</td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td>features:</td>
<td>Power steering, Air conditioning..</td>
</tr>
<tr>
<td>description:</td>
<td>COMMODORE VP, '93 Series 11, auto., air, steer, $5200 ono. YNR-126, PH. 4959 1992</td>
</tr>
<tr>
<td>location:</td>
<td>The Newcastle Herald and Post</td>
</tr>
</tbody>
</table>

### CAR

<table>
<thead>
<tr>
<th>Unknown1:</th>
<th>HOLDEN COMMODORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown2:</td>
<td>$7250</td>
</tr>
<tr>
<td>year:</td>
<td>1992</td>
</tr>
<tr>
<td>kms:</td>
<td>175000</td>
</tr>
<tr>
<td>trans:</td>
<td></td>
</tr>
<tr>
<td>body:</td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td>features:</td>
<td></td>
</tr>
<tr>
<td>description:</td>
<td></td>
</tr>
<tr>
<td>location:</td>
<td></td>
</tr>
</tbody>
</table>

### CAR

<table>
<thead>
<tr>
<th>Unknown1:</th>
<th>TOYOTA CAMRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown2:</td>
<td>$6500</td>
</tr>
<tr>
<td>year:</td>
<td>1991</td>
</tr>
<tr>
<td>kms:</td>
<td>53000</td>
</tr>
<tr>
<td>trans:</td>
<td>A</td>
</tr>
<tr>
<td>body:</td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td>features:</td>
<td></td>
</tr>
<tr>
<td>description:</td>
<td></td>
</tr>
<tr>
<td>location:</td>
<td></td>
</tr>
</tbody>
</table>

## VEHICLE

<table>
<thead>
<tr>
<th>BIDS:</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR:</td>
<td>Ford Fairlane 1991 NC</td>
</tr>
<tr>
<td>PRICE:</td>
<td>AU $6,000.00</td>
</tr>
<tr>
<td>TIME_LEFT:</td>
<td>4d 02h 33m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS:</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR:</td>
<td>MITSUBISHI MAGNA TJ SPORTS 2001 MODEL</td>
</tr>
<tr>
<td>PRICE:</td>
<td>AU $25,000.00</td>
</tr>
<tr>
<td>TIME_LEFT:</td>
<td>6d 13h 30m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS:</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR:</td>
<td>MUST BE SEEN. SERIES II VR SS V8 5 SPEED</td>
</tr>
<tr>
<td>PRICE:</td>
<td>AU $12,100.00</td>
</tr>
<tr>
<td>TIME_LEFT:</td>
<td>17h 30m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS:</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR:</td>
<td>BMW E30 Series John Player Special 323i Coupe</td>
</tr>
<tr>
<td>PRICE:</td>
<td>AU $10,950.00</td>
</tr>
<tr>
<td>TIME_LEFT:</td>
<td>9d 16h 51m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS:</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR:</td>
<td>FORD Laser KF TX3 1.8 ltr - 1990</td>
</tr>
<tr>
<td>PRICE:</td>
<td>AU $2,560.00</td>
</tr>
<tr>
<td>TIME_LEFT:</td>
<td>2d 19h 02m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS:</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR:</td>
<td>Mitsubishi L300 van</td>
</tr>
<tr>
<td>PRICE:</td>
<td>AU $1,000.00</td>
</tr>
<tr>
<td>TIME_LEFT:</td>
<td>2d 01h 14m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS:</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR:</td>
<td>69 Camaro 396 RS/SS Matching # car</td>
</tr>
<tr>
<td>PRICE:</td>
<td>AU $40,000.00</td>
</tr>
<tr>
<td>TIME_LEFT:</td>
<td>5d 00h 29m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS:</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR:</td>
<td>fuel saver economy increases lowers emissions</td>
</tr>
<tr>
<td>PRICE:</td>
<td>AU $35.00</td>
</tr>
<tr>
<td>TIME_LEFT:</td>
<td>6d 01h 56m</td>
</tr>
<tr>
<td>eBay</td>
<td>Autotrader</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MAKE_MODEL: Mazda 323 Astina</td>
<td></td>
</tr>
<tr>
<td>YR: 1992</td>
<td></td>
</tr>
<tr>
<td>KMS: 140,000</td>
<td></td>
</tr>
<tr>
<td>PRICE: $9,999</td>
<td></td>
</tr>
<tr>
<td>STATE: QLD</td>
<td></td>
</tr>
<tr>
<td>DEALER: Ferrierr Qualit</td>
<td></td>
</tr>
<tr>
<td>MAKE_MODEL: Saab 9000</td>
<td></td>
</tr>
<tr>
<td>YR: 1987</td>
<td></td>
</tr>
<tr>
<td>KMS:</td>
<td></td>
</tr>
<tr>
<td>PRICE: $5,990</td>
<td></td>
</tr>
<tr>
<td>STATE: SA</td>
<td></td>
</tr>
<tr>
<td>DEALER: Auto Credit Co</td>
<td></td>
</tr>
<tr>
<td>MAKE_MODEL: Subaru Liberty GX</td>
<td></td>
</tr>
<tr>
<td>YR: 1993</td>
<td></td>
</tr>
<tr>
<td>KMS:</td>
<td></td>
</tr>
<tr>
<td>PRICE: $9,990</td>
<td></td>
</tr>
<tr>
<td>STATE: VIC</td>
<td></td>
</tr>
<tr>
<td>DEALER: Eastern Vehicle</td>
<td></td>
</tr>
<tr>
<td>MAKE_MODEL: Ford Falcon XH</td>
<td></td>
</tr>
<tr>
<td>YR: 1997</td>
<td></td>
</tr>
<tr>
<td>KMS:</td>
<td></td>
</tr>
<tr>
<td>PRICE: $9,999</td>
<td></td>
</tr>
<tr>
<td>STATE: VIC</td>
<td></td>
</tr>
<tr>
<td>DEALER: Lucky Choice Ca</td>
<td></td>
</tr>
<tr>
<td>MAKE_MODEL: Ford Fairlane</td>
<td></td>
</tr>
<tr>
<td>YR: 1991</td>
<td></td>
</tr>
<tr>
<td>KMS: 176,330</td>
<td></td>
</tr>
<tr>
<td>PRICE: $9,000</td>
<td></td>
</tr>
<tr>
<td>STATE: NSW</td>
<td></td>
</tr>
<tr>
<td>DEALER:</td>
<td></td>
</tr>
<tr>
<td>MAKE_MODEL: Mitsubishi Magna</td>
<td></td>
</tr>
<tr>
<td>YR: 1994</td>
<td></td>
</tr>
<tr>
<td>KMS: 142,000</td>
<td></td>
</tr>
<tr>
<td>PRICE: $8,990</td>
<td></td>
</tr>
<tr>
<td>STATE: NSW</td>
<td></td>
</tr>
<tr>
<td>DEALER: Laurieton Motor</td>
<td></td>
</tr>
</tbody>
</table>
eBay vs AutoWeb

<table>
<thead>
<tr>
<th>VEHICLE</th>
<th>BIDS: 0</th>
<th>CAR: Ford Fairlane 1991 NC</th>
<th>PRICE: AU $6,000.00</th>
<th>TIME_LEFT: 4d 02h 33m</th>
</tr>
</thead>
<tbody>
<tr>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: MITSUBISHI MAGNA TJ SPORTS 2001 MODEL</td>
<td>PRICE: AU $25,000.00</td>
<td>TIME_LEFT: 6d 13h 30m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 5</td>
<td>CAR: MUST BE SEEN. SERIES II VR SS V8 5 SPEED</td>
<td>PRICE: AU $12,100.00</td>
<td>TIME_LEFT: 17h 30m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: BMW E30 Series John Player Special 323i Coupe</td>
<td>PRICE: AU $10,950.00</td>
<td>TIME_LEFT: 9d 16h 51m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: 1985 Holden VK Commodore Executive Unreg</td>
<td>PRICE: AU $500.00</td>
<td>TIME_LEFT: 6d 14h 12m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 10</td>
<td>CAR: FORD Laser KF TX3 1.8 ltr - 1990</td>
<td>PRICE: AU $2,560.00</td>
<td>TIME_LEFT: 2d 19h 02m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 1</td>
<td>CAR: Mitsubishi L300 van</td>
<td>PRICE: AU $1,000.00</td>
<td>TIME_LEFT: 2d 01h 14m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: 69 Camaro 396 RS/SS Matching # car</td>
<td>PRICE: AU $40,000.00</td>
<td>TIME_LEFT: 5d 00h 29m</td>
</tr>
<tr>
<td>VEHICLE</td>
<td>BIDS: 0</td>
<td>CAR: fuel saver economy increases lowers emissions</td>
<td>PRICE: AU $35.00</td>
<td>TIME_LEFT: 6d 01h 56m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VEHICLE</td>
<td>YEAR: 1996</td>
<td>DESCRIPTION: MITSUBISHI LANCIER GLICE</td>
<td>PRICE: $9,990</td>
<td>BODY_TYPE: COUPE</td>
<td>COLOUR: METALLIC BLUE</td>
<td>LOCATION:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>eBay</th>
<th>AutoWeb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Confidence (%)
### eBay vs Carsales

<table>
<thead>
<tr>
<th>VEHICLE</th>
<th>BIDS</th>
<th>CAR</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>Ford Fairlane 1991 NC</td>
<td>AU $6,000.00</td>
<td>4d 02h 33m</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>MITSUBISHI MAGNA TJ SPORTS 2001 MODEL</td>
<td>AU $25,000.00</td>
<td>6d 13h 30m</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>MUST BE SEEN. SERIES II VR SS V8 5 SPEED</td>
<td>AU $12,100.00</td>
<td>2d 19h 02m</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1985 Holden VK Commodore Executive Unreg</td>
<td>AU $500.00</td>
<td>6d 14h 12m</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>FORD Laser KF TX3 1.8 ltr - 1990</td>
<td>AU $2,560.00</td>
<td>2d 19h 02m</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Mitsubishi L300 van</td>
<td>AU $1,000.00</td>
<td>2d 01h 14m</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>69 Camaro 396 RS/SS Matching # car</td>
<td>AU $40,000.00</td>
<td>5d 00h 29m</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>fuel saver economy increases lowers emissions</td>
<td>AU $35.00</td>
<td>6d 01h 56m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VEHICLE</th>
<th>BIDS</th>
<th>CAR</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BMW E30 Series John Player Special 323i Coupe</td>
<td>AU $10,950.00</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VEHICLE</th>
<th>BIDS</th>
<th>CAR</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>1985 Holden VK Commodore Executive Unreg</td>
<td>AU $500.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>FORD Laser KF TX3 1.8 ltr - 1990</td>
<td>AU $2,560.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Mitsubishi L300 van</td>
<td>AU $1,000.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>69 Camaro 396 RS/SS Matching # car</td>
<td>AU $40,000.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>fuel saver economy increases lowers emissions</td>
<td>AU $35.00</td>
<td></td>
</tr>
</tbody>
</table>

| CAR | Unknown1: | 1986 MAZDA RX7 FC1031 Limited, 2 door COUPE, white, 4 sp Automatic, 170000km, Petrol Rotary 1.3 2cyl (1308cc). Reg. No UDK-569 |
|     | Unknown2: | $6,500 |
|     | Unknown3: | Only $6500 ONO. price is negotiable. the 5th picture is what the car COULD look like. |
|     | Unknown4: | Fulham Gardens, Adelaide, SA |

| CAR | Unknown1: | 1993 FORD LASER KH S, 5 door HATCHBACK, green, 5 sp Manual, 245000km, Petrol Multi-point injected 1.8 4cyl (1840cc). Reg. No NBS 437 |
|     | Unknown2: | $6,000 |
|     | Unknown3: | laser 93KH sports,1.8 EFI, air cond,p-steer,t-windows, $6000 |
|     | Unknown4: | Wodonga, VIC |

| CAR | Unknown1: | 1997 PROTON SATRIA GLi, 3 door HATCHBACK, GREEN, 5 sp Manual, 112000km, Petrol Multi-point injected 1.5 4cyl (1468cc). Reg. No UNH-993 |
|     | Unknown2: | $5,950 |
|     | Unknown3: | Good clean economical car - priced to sell |
|     | Unknown4: | Castle Hill, NSW |

| CAR | Unknown1: | 1984 MERCEDES 230E W123, 4 door SEDAN, Champagne (gold), 4 sp Automatic, 220000km. Petrol Multi-point injected 2.3 4cyl (2299cc). Reg. No OIL363 |
|     | Unknown2: | $6,500 |
|     | Unknown3: | 6 seater |
|     | Unknown4: | Warner Brisbane, QLD |

| CAR | Unknown1: | 1992 FORD FALCON EB II GLi, 4 door WAGON, red, 4 sp Automatic, 200000km, Petrol Multi-point injected 4.0 6cyl (3984cc). Reg. No 415GTC |
|     | Unknown2: | $5,100 |
|     | Unknown3: | 6 seater |
|     | Unknown4: | Warner Brisbane, QLD |

<table>
<thead>
<tr>
<th>eBay</th>
<th>Carsales</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
eBay vs Drive

<table>
<thead>
<tr>
<th>eBay</th>
<th>Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Confidence (%)**

<table>
<thead>
<tr>
<th>eBay</th>
<th>Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**VEHICLE**

<table>
<thead>
<tr>
<th>BIDS</th>
<th>CAR:</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$6,000.00</td>
<td>4d 02h 33m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS</th>
<th>CAR:</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MITSUBISHI MAGNA TJ SPORTS 2001 MODEL</td>
<td>$25,000.00</td>
<td>6d 13h 30m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS</th>
<th>CAR:</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MUST BE SEEN. SERIES II VR SS V8 5 SPEED</td>
<td>$12,100.00</td>
<td>17h 30m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS</th>
<th>CAR:</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BMW E30 Series</td>
<td>$10,950.00</td>
<td>9d 16h 51m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS</th>
<th>CAR:</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1985 Holden VK Commodore Executive</td>
<td>$500.00</td>
<td>6d 14h 12m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS</th>
<th>CAR:</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FORD Laser KF TX3 1.8 ltr - 1990</td>
<td>$2,560.00</td>
<td>2d 19h 02m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIDS</th>
<th>CAR:</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1985 Camaro 396 RS/SS Matching</td>
<td>$35.00</td>
<td>6d 01h 56m</td>
</tr>
</tbody>
</table>

**CAR**

<table>
<thead>
<tr>
<th>Unknown1:</th>
<th>MERCEDES-BENZ 380</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown2:</td>
<td>$9500</td>
</tr>
<tr>
<td>year:</td>
<td>0</td>
</tr>
<tr>
<td>kms:</td>
<td></td>
</tr>
<tr>
<td>trans:</td>
<td></td>
</tr>
<tr>
<td>body:</td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td>features:</td>
<td>Anti lock braking, Alarm system, Cruise control, Radio cassette...</td>
</tr>
<tr>
<td>description:</td>
<td>MERCEDES 380SEL, Diamond Blue, blue lther, recond motor, Alpine stereo, alarm, ABS, sunrf, cruise, reg 1/04 YDQ756. $9500 ono 0439 311 767.</td>
</tr>
<tr>
<td>location:</td>
<td>The Sydney Morning Herald</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unknown1:</th>
<th>HOLDEN COMMODORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown2:</td>
<td>$5200</td>
</tr>
<tr>
<td>year:</td>
<td>1993</td>
</tr>
<tr>
<td>kms:</td>
<td>175000</td>
</tr>
<tr>
<td>trans:</td>
<td>A</td>
</tr>
<tr>
<td>body:</td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td>features:</td>
<td>Power steering, Air conditioning...</td>
</tr>
<tr>
<td>description:</td>
<td>COMMODORE VP, '93 Series 11, auto., air, steer, $5200 ono. YNR-126. PH. 4959 1992</td>
</tr>
<tr>
<td>location:</td>
<td>The Newcastle Herald and Post</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unknown1:</th>
<th>HOLDEN COMMODORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown2:</td>
<td>$7250</td>
</tr>
<tr>
<td>year:</td>
<td>1992</td>
</tr>
<tr>
<td>kms:</td>
<td></td>
</tr>
<tr>
<td>trans:</td>
<td></td>
</tr>
<tr>
<td>body:</td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td>features:</td>
<td></td>
</tr>
<tr>
<td>description:</td>
<td></td>
</tr>
<tr>
<td>location:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unknown1:</th>
<th>TOYOTA CAMRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown2:</td>
<td>$6500</td>
</tr>
<tr>
<td>year:</td>
<td>1991</td>
</tr>
<tr>
<td>kms:</td>
<td>53000</td>
</tr>
<tr>
<td>trans:</td>
<td>A</td>
</tr>
<tr>
<td>body:</td>
<td></td>
</tr>
<tr>
<td>colour:</td>
<td></td>
</tr>
<tr>
<td>ad_date:</td>
<td>03/05/2003</td>
</tr>
<tr>
<td>features:</td>
<td></td>
</tr>
<tr>
<td>description:</td>
<td></td>
</tr>
<tr>
<td>location:</td>
<td>The Newcastle Herald and Post</td>
</tr>
</tbody>
</table>

**Fuel saver economy increases lowers emissions**

**Location**

- The Sydney Morning Herald
- The Newcastle Herald and Post
- The Newcastle Herald and Post

**Description**

- MERCEDES 380SEL, Diamond Blue, blue lther, recond motor, Alpine stereo, alarm, ABS, sunrf, cruise, reg 1/04 YDQ756. $9500 ono 0439 311 767.
- COMMODORE VP, '93 Series 11, auto., air, steer, $5200 ono. YNR-126. PH. 4959 1992
- COMMODORE VP, '92 Series II, TSU-387, regd Nov '93, new paint, tyres, very good cond, 175,000 ks, $7250 ono. 4965 9112.
- TOYOTA CAMRY '91 EXECUTIVE AUTO ONE OWNER 53,000 k's $6500. RWK-150. PH: 4946 5548
<table>
<thead>
<tr>
<th>VEHICLE</th>
<th>BIDS</th>
<th>CAR</th>
<th>PRICE</th>
<th>TIME_LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>eBay vs eBay</td>
<td>0</td>
<td>Ford Fairlane 1991 NC</td>
<td>AU $6,000.00</td>
<td>4d 02h 33m</td>
</tr>
<tr>
<td>eBay vs eBay</td>
<td>0</td>
<td>MITSUBISHI MAGNA TJ SPORTS 2001 MODEL</td>
<td>AU $25,000.00</td>
<td>6d 13h 30m</td>
</tr>
<tr>
<td>eBay vs eBay</td>
<td>5</td>
<td>MUST BE SEEN. SERIES II VR SS V8 5 SPEED</td>
<td>AU $12,100.00</td>
<td>17h 30m</td>
</tr>
<tr>
<td>eBay vs eBay</td>
<td>0</td>
<td>BMW E30 Series John Player Special 323i Coupe</td>
<td>AU $10,950.00</td>
<td>9d 16h 51m</td>
</tr>
<tr>
<td>eBay vs eBay</td>
<td>0</td>
<td>1985 Holden VK Commodore Executive Unreg</td>
<td>AU $500.00</td>
<td>6d 14h 12m</td>
</tr>
<tr>
<td>eBay vs eBay</td>
<td>10</td>
<td>FORD Laser KF TX3 1.8 ltr - 1990</td>
<td>AU $2,560.00</td>
<td>2d 19h 02m</td>
</tr>
<tr>
<td>eBay vs eBay</td>
<td>1</td>
<td>Mitsubishi L300 van</td>
<td>AU $1,000.00</td>
<td>2d 01h 14m</td>
</tr>
<tr>
<td>eBay vs eBay</td>
<td>0</td>
<td>69 Camaro 396 RS/SS Matching # car</td>
<td>AU $40,000.00</td>
<td>5d 00h 29m</td>
</tr>
<tr>
<td>eBay vs eBay</td>
<td>0</td>
<td>fuel saver economy increases lowers emissions</td>
<td>AU $35.00</td>
<td>6d 01h 56m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>eBay</th>
<th>eBay</th>
<th>Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **VEHICLE**
- **BIDS**
- **CAR**
- **PRICE**
- **TIME_LEFT**
Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Lister, Kendall

Title:
Toward semantic interoperability for software systems

Date:
2008

Citation:

Persistent Link:
http://hdl.handle.net/11343/35107

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.