ESTRELLA
IST-2004-027655

European project for Standardized Transparent Representations in order to Extend LegaL Accessibility Specific Targeted Research or Innovation Project

Specific Targeted Research Project
Information Society Technologies

Deliverable N°: 4.5

ESTRELLA user report

Version:	FINAL 1.0
Due date of Deliverable:	September 30, 2008
Actual submission date:	September 10, 2008
Start date of Project:	1 January 2006
Duration:	30 months
Project Coordinator:	Universiteit van Amsterdam (NL)
Lead contractor deliverable:	University of Liverpool
Participating contractors:	Alma Mater Studiorum - Universita di Bologna (IT), University of Liverpool (UK), Fraunhofer Gesellschaft zur foerderung der angewandten forschung e.v. (DE), RuleWise b.v. (NL), RuleBurst (EUROPE) Ltd. (UK), knowledgeTools International GmbH (DE), Interaction Design Ltd. (UK), SOGEI - Societa Generale d'Informatica S.P.A. (IT), Ministro per le Riforme e le Innivazioni nella Publica Amministrazione (IT), Hungarian Tax and Financial Control Administration (HU), Budapesti Corvinus Egyetem (HU), Ministero dell'Economia e delle Finanze (IT), Consorzio Pisa Ricerche SCARL (IT)

Project funded by the European Community under the 6th Framework Programme

	Dissemination Level	
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Executive Summary

The European project for Standardized Transparent Representations in order to Extend Legal Accessibility (Estrella, IST-2004-027655) has developed and validated an open, standards-based platform which allows public administrations to develop and deploy comprehensive legal knowledge management solutions without becoming dependent on proprietary products of particular vendors. The main technical achievement of the project is the Legal Knowledge Interchange Format (LKIF), which represents legal information in a form which builds upon emerging XML-based standards of the Semantic Web. The Estrella User Report provides an overview of the objectives and outcomes of the Estrella Project in order to promote adoption of the platform by organisations which would benefit from the capabilities of the project's deliverables.

In the first and second chapters, we introduce the report, then provide an overview of the document and briefly discuss the background context of the Estrella Project. The purposes of the project are outlined in chapter three. In chapter four, we justify the design choices in LKIF, including discussions about ontologies, the Semantic Web, defeasible rules, hybrid reasoning, argument visualisation, application program interfaces, an inference engine, and document management. The research and results in the project are set in the context of research in artificial intelligence and the law. The fifth chapter presents brief reports from project participants about the translators which they developed to translate between their representation of legal knowledge and LKIF, demonstrating the viability and usefulness of the results of the project. The final chapter outlines pilot studies by project participants in which legal documents were represented and subsequently translated.

Estrella User Report
Deliverable 4.5

Adam Wyner and Trevor Bench-Capon
University of Liverpool

The European project for Standardized Transparent Representations in order to Extend Legal Accessibility (Estrella Project) has developed and validated an open, standards-based platform which allows public administrations to develop and deploy comprehensive legal knowledge management solutions without becoming dependent on proprietary products of particular vendors. The main technical achievement of the project is the Legal Knowledge Interchange Format (LKIF), which represents legal information in a form which builds upon emerging XML-based standards of the Semantic Web. This report is an overview of the objectives and outcomes of the project in order to promote adoption of the platform by organisations which would benefit from the capabilities of the project's deliverables.
The report is first introduced, providing an overview of the document and the background context of the Estrella Project. The purposes of the project are outlined. The design choices of LKIF are justified, including discussions about ontologies, the Semantic Web, defeasible rules, hybrid reasoning, argument visualisation, application program interfaces, an inference engine, and document management. The research and results in the project are set in the context of research in artificial intelligence and the law. In the final two chapters, the project participants first report on their translators between their representation of legal knowledge and LKIF, then outline pilot studies in which legal documents were represented and subsequently translated.

Contents

ESTRELLA
p/a Faculty of Law
University of Amsterdam
PO Box 1030
1000BA Amsterdam
The Netherlands

tel: +31 20 525 3485/3490
fax: +31 20 525 3495
http://www.estrellaproject.org

Corresponding author:
Adam Wyner
tel: +44 (0)151 795 4294
adam@wyner.info

Chapter 1

Introduction

The European project for Standardized Transparent Representations in order to Extend Legal Accessibility (Estrella Project, IST-2004-027655) has developed and validated an open, standards-based platform which allows public administrations to develop and deploy comprehensive legal knowledge management solutions without becoming dependent on proprietary products of particular vendors. Estrella supports a complete and integrated system for legal document management and legal knowledge-based representation and reasoning. The system can help to improve the quality and efficiency of the determinative processes of public administration which make use of complex legislation and a range of legal sources. As an open, standards-based platform, Estrella facilitates a market of interoperable components for legal document management and knowledge-based systems. Thus, public administrations and other users can freely choose among competing development environments, inference engines, and other tools.

The main technical achievement of the Estrella project is the Legal Knowledge Interchange Format (LKIF), building upon emerging XML-based standards of the Semantic Web, including RDF and OWL, along with Application Programmer Interfaces (APIs) for interacting with LKIF.

To achieve vendor neutrality and independence, Estrella provides translators between the LKIF format and the existing proprietary formats of participating vendors. To demonstrate and validate the Estrella platform, portions of European tax related legislation and national tax legislation of some European countries have been modelled and used in the pilot applications. The finance ministries or tax administrations of several European countries have participated on an Observatory Board so as to ensure the generality of the approach.

This document introduces and outlines the main purposes, achievements, and deliverables of the Estrella Project. In the following, we briefly discuss the background context of the Estrella Project, then outline the purposes of LKIF in more detail, justifying its design features. Work by vendors on translation and pilot projects are briefly outlined. In addition, the report discusses related work that was also developed in the Estrella Project: an ontology of basic legal concepts and a document management system for legal sources.

We should emphasise that this document is a user report. For a complete reference to all the features, tutorial, technical information, and examples of the Estrella Platform, one may consult deliverables that are available from the Estrella Project:

http://www.estrellaproject.org

Chapter 2

Overview and Background Context

In this chapter, we provide an overview of the Estrella Project and the background context in which the project was initiated. In section 4.2, we show that the Estrella Project is well-founded in research on artificial intelligence and the law.

2.1 Overview

The Estrella Project (The European project for Standardized Transparent Representations in order to Extend Legal Accessibility) has developed a platform which allows public administrations to deploy comprehensive solutions for the management of legal knowledge. In reasoning about social benefits or taxation, public administrators must represent and reason with complex legislation. The platform is intended to support the representation of and reasoning about legislation in a way that can help public administrations to improve the quality and efficiency of their services. Moreover, given a suitable interface, the legislation can be made available for the public to interact with. For example, LKIF tools could be made available to citizens via the web to help them to assess their eligibility for a social benefits as well as filling out the appropriate application forms.

The platform has been designed to be open and standardised so that public administrations need not become dependent on proprietary products of particular vendors. Along the same lines, the platform supports interoperability among various components for legal knowledge-based systems allowing public administrations to freely choose among the components. A standardised platform also enables a range of vendors to develop innovative products to suit particular market needs without having to be concerned with an all-encompassing solution, compatibility with other vendors, or being locked out of a strategic market by “monolithic” vendors. As well, the platform abstracts from the expression of legislation in different natural languages so providing a common, abstract legal “lingua franca”.

The main technical achievement of the Estrella Project is the development of a Legal Knowledge Interchange Format (LKIF), which represents legal information in a form which builds upon emerging XML-based standards of the Semantic Web. The project platform provides Application Programmer Interfaces (APIs) for interacting with legal knowledge-based systems using LKIF. LKIF provides formalisms for representing concepts (“ontologies”), inference rules, precedent cases and arguments. An XML document schema for legislation has been developed, called MetaLex, which complements and integrates national XML standards for legislation. This format supports document search, exchange, and association among documents as well as enforces a link between legal sources and the legal knowledge systems which reason about the information in the sources. In addition, a reference inference engine has been developed which supports reasoning with legal knowledge represented in LKIF. The utility of LKIF as an interchange format for legal knowledge has been demonstrated with pilot tests of legal documents which are expressed in proprietary formats of several vendors then translated to and from the format of one vendor to that of another.

The Estrella Project was formed in the Fall of 2006 as a consortium comprised of 15 partners drawn from companies, non-profit research institutes, universities, and public administrations in five EU countries (see Appendix). The project was completed in the Fall of 2008.

2.2 Background Context

The Estrella Project originated in the context of European Union integration, where:

- The European Parliament passes EU wide directives which need to be incorporated into or related to the legislation of member states.
- Goods, services, and citizens are free to move across open European borders.
- Democratic institutions must be strengthened as well as be more responsive to the will of the citizenry.
- Public administrations must be more efficient and economical.

In the EU, the legal systems of member states have been composed of heterogeneous, often conflicting, rules and regulations concerning taxes, employment, education, pensions, health care, property, trade, and so on. Integration of new EU legislation with existing legislation of the member states as well as homogenisation of legal systems across the EU has been problematic, complex, and expensive to implement. As the borders of member states open, the rules and regulations concerning the benefits and liabilities of citizens and businesses must move as people, goods, and services move. For example, laws concerning employment and pension ought to be comparable across the member states so as to facilitate the movement of employees across national boundaries. In addition, there are more general concerns about improving the functionality of the legal system so as to garner public support for the legal system, promoting transparency, compliance, and citizen involvement. Finally, the costs of administering the legal system by EU administrative departments, administrations of member states, and companies throughout the EU are significant and rising. The more complex and dynamic the legislative environment, the more burdensome the costs.

Chapter 3

Purposes

Given this background context, the Estrella Project was initiated with the following purposes in mind:

- to facilitate the integration of EU legal systems
- to modernise public administration at the levels of the EU and within member states by supporting efficiency, transparency, accountability, accessibility, inclusiveness, portability, and simplicity of core governmental processes and services
- to improve the quality of legal information by testing legal systems for consistency (are there contradictions between portions of the law) and correctness (is the law achieving the goal it is specified for?).
- to reduce the costs of public administration
- to reduce private sector costs of managing their legal obligations
- to encourage public support for democratic institutions by participation, transparency, and personalisation of services
- to ease the mobility of goods, services, and EU citizens within the EU
- to support businesses across EU member states
- to provide the means to "modularise" the legal systems for different levels of EU legal structure, e.g. provide a "municipal government" module which could be amended to suit local circumstances
- to support a range of governmental and legal processes across organisations and on behalf of citizens and businesses
- to support a variety of reasoning patterns as needed across a range of resources (e.g. directives, legal case bases).

Chapter 4

Justification of Design Choices

In this chapter, we justify the choices which appear in the design of the components of the platform – LKIF, the inference engine, and the document management system. The justifications are grounded in *what* knowledge is to be represented and *how* this knowledge is to be processed.

4.1 LKIF

The design of LKIF is guided by the following main considerations, which we discuss further below ([Estrella Project, 2007] and [Gordon, 2008a]).

- *Legal knowledge* must be represented across domains such as tax, social security, business, penal, and so on. Legal concepts, rules, cases and arguments must be represented.
- It must be an *interchange format*.
- It should reuse Semantic Web *XML-based standards* when applicable.
- Application Program Interfaces (APIs) must be available to support the use of LKIF.
- The system should be informed by the state-of-the-art of research in the field of artificial intelligence and the law.

Legal knowledge here means the concepts of jurisprudence as well as the knowledge and practice of the law according to lawyers, judges, and legislators. Legal knowledge is largely encoded in the texts in which the law is expressed, namely texts on jurisprudence, legislative acts, legal instruments such as contracts, and case law. In addition, people who practice the law or civil servants who administer the law know about the organisation of legal documents or about the procedures that are applied when working with the law. *Argumentation* is another aspect of legal knowledge, for lawyers, judges, legislators, and civil servants argue about the meaning of legislation, cases, and the facts which are supported by the evidence in a given case. This includes issues relating to priorities among and exceptions to legal rules. In addition, the representation of legal knowledge must be accessible to the populus served by the law. In determining the scope of this knowledge, the Estrella Project has relied on scientists who conduct research in the field of artificial intelligence and law, on partners from government ministries who are experts in particular legal domains (e.g. tax legislation), and on experiments in modeling particular pieces of legislation. As a range of legal knowledge is considered, different approaches were melded into a cohesive framework.

4.1.1 Ontological Modeling

One approach is to model the terminology in an ontology. Some aspects of legal knowledge can be represented in an ontology, where one gives definitions and properties of as well as relationships among terminology which appears in legislation. However, as we discuss below, other aspects of legal knowledge cannot be represented in an ontology. An ontology defines the terms that are used to describe and represent the concepts of an area of knowledge as well as the relationships between the terms. The concepts are described in a taxonomy with respect to which necessary and/or sufficient conditions which hold. An ontology is not only useful as an explicit specification of domain knowledge, but also allows one to query the knowledge base and draw inferences. For example, an ontology to describe and represent a family would include concepts such as "parent of", "daughter of", as well as "brother", "mother", and others. Given two individuals with the same parent, we can infer relative to the ontology that the individuals are siblings. Given some ontology and instances which satisfy the properties of the ontology, one may query the knowledge base, e.g. asking for all the children of a particular individual returns those individuals who are children.

Ontologies may be provided for any sort of domain of knowledge. For the legal domain, it was determined that concepts which appear in legal documents would be represented in the ontology to the extent possible. Consequently, the *foundational* component of the ontology represents general conceptual knowledge about space, time, actions, and physical objects as well as mental and social concepts. More specific to the legal domain, the *legal core ontology* represents general legal concepts such as legal actions, codes, persons, norms, and judicial organisations. This ontology is further refined to a *legal domain ontology* with respect to particular legal codes to represent, for example, particular criminal acts, individuals bearing responsibility, particular court structures in a legal system. Finally, an ontology to support case-based reasoning is provided so as to represent the parts and relationships among cases in a case base [Wyner, 2007].

4.1.2 Ontology Web Language

In addition to these legal requirements, the design of LKIF should make use of emerging World Wide Web standards for the Semantic Web, where feasible, to leverage existing tools, infrastructure, and expertise. For this, the *Ontology Web Language* (OWL), a World Wide Web standard, was chosen for the representation of terminological knowledge. It has a well-defined syntax and semantics along with a range of tools for developing an ontology. In addition, a document marked up with the properties and relations relative to an ontology can be indexed and searched.

The project keeps in mind computational issues related to the expressivity of the ontology and to support tools for representing and reasoning with the ontology. There exist a variety of tools for developing and using OWL ontologies, such as the Protege ontology editor:

`http://protege.stanford.edu/`

as well as the Pellet inference engine:

`http://pellet.owldl.com/`

4.1.3 Defeasible Rules, Arguments, and Cases

While an existing Web standard, OWL was able to be adopted for use in LKIF to represent legal concepts, no existing standard for rules was found to be adequate for modeling *legal rules*, *legal arguments*, or *case-based reasoning*, nor did existing standards reflect the state-of-the-art in the field of artificial intelligence and law ([Gordon et al., 2007a], [Gordon, 2007b], [Gordon, 2007a], [Gordon, 2008b]).

Legal rules are *defeasible*, meaning that the conclusions of the rules are only presumptively true, rather than necessarily true, when their conditions are satisfied. In contrast, in classical logical systems, if the conditions are satisfied, the conclusion *must* follow. Defeasible rules can be subject to exceptions or defeated in various ways, say by conflicting rules of higher priority, which requires the representation of meta-level information such as the date of enactment of a rule. For example, a pensioner (given a suitable definition) may be eligible for benefits if certain conditions are met except when certain other conditions hold as well. Furthermore, while classical logical systems must explicitly represent *all* assumptions, natural reasoning leaves some *implicit*; the reasoning system must help users to reveal implicit assumptions during the analysis of a case. In addition, legal rules are subject to exclusions, which are statements of when a rule is inapplicable. Thus, in general, a formalism for legal rules must be capable of representing priorities, negation, exceptions, assumptions, and exclusion.

Lawyers *argue* about the pros and cons of cases with respect to evidence and legal rules; legal cases are decided citing the arguments and providing explanations for the outcome.

In order to support *case-based reasoning*, the knowledge representation language provides a way to represent the factors and decisions of precedent cases as well as to reason with them [Wyner and Bench-Capon, 2007].

4.1.4 Hybrid Reasoning

Because legal reasoning requires the construction and comparison of various kinds of arguments, a *hybrid reasoning* approach was adopted. In this approach, LKIF knowledge bases are used with argumentation patterns, called argumentation schemes, to construct arguments from the concepts, rules, and cases which are modeled in the knowledge base. Argumentation schemes represent basic reasoning patterns and are defeasible rules. For example, the argument scheme *Argument from Position to Know*:

Position to Know Premise: Person p is in a position to know whether the statement s is true of false.

Assertion Premise: p asserts that s is true (or false).

Conclusion: s is true (or false).

The conclusion presumptively follows from the premises. However, one may attack whether or not the person is in a position to know. A variety of other argumentation schemes are available for reasoning about ontologies and cases. Argument schemes are *instantiated* and inter-related in relations of attack and support, producing an *argument graph*. An inference engine has been designed to reason with argument schemes; in addition, the engine provides a *decision procedure* for determining whether a given claim is presumably true relative to the argument graph. The output is a graphical representation of the arguments pro and con the claim.

4.1.5 LKIF

The Legal Knowledge Interchange Format (LKIF), is a Semantic Web-based language for modeling legal domains and to facilitate interchange between legal knowledge-based systems. A language which suits the Semantic Web uses XML-based standards; legal information is represented using an XML-based language wherein the components of information and their relationships are explicitly marked in a file and in such a way as to support search, automated manipulation, and inference on a local machine as well as remotely over the internet. That LKIF is an interchange format means that it is a high-level, generic, and vendor neutral representation of legal knowledge that is not tied to particular representation languages of any vendor or to the processes that they apply to that language. In turn, this is motivated by the purpose of providing an open, standardised platform, which has several advantages outlined above. The success of an interchange format is justified where several vendors with proprietary representational formats and processes can translate into and out of the interchange format, thus sharing a core, common knowledge base.

One of the main objectives of LKIF as an interchange language is to enable legal reasoning across a range of technologies. For example, *expert systems* are a widespread and well-developed approach to rule-based reasoning with respect to a knowledge-base. An alternative approach is *logic trees*, which represent reasoning in *AND/OR* graphs. LKIF is designed to translate into and out of these alternative representation and reasoning systems, which we discuss further in chapter 5.

4.1.6 Argument Visualisation

An additional benefit of an XML based language for the representation of knowledge, rules, and arguments is that the information can alternatively be represented *graphically*, which allows users to easily grasp the nature of the statements and their relationships.

Argument visualisation has a long history in the legal profession and is of current interest ([Wigmore, 1931], [Kirschner et al., 2003]). Carneades has an allied graphic

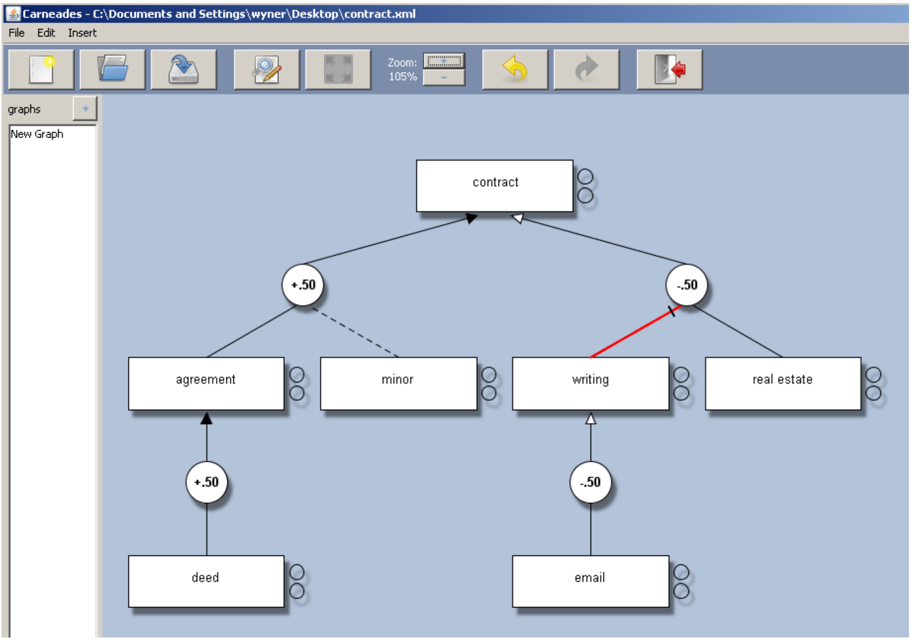

Figure 4.1: A Representation of a Contract in Carneades

representation in which premises (of various sorts) and conclusions are not only represented, but can be reasoned with in the graph by indicating which statements are accepted, rejected, or have other values. Given the input values, Carneades calculates the result and indicates this in the graphic.

In Figure 4.1, we have a sample graphic representation of a contract in a graphic representation in Carneades. The interpretation is:

> There is a contract if there is an agreement except with a minor. There is an agreement if there is a deed. There is not a contract if there is real estate and if the agreement is not in writing. The agreement is not in writing if it is an email.

The black arrowheads indicate *pro* premises for the conclusion, while the white arrowhead indicate *con* premises against the conclusion. Dashed lines represent exceptions. The pro and con premises can be *weighted* for their effect (1.0-0.0).

To build a graph, one enters a statement which is a premise or conclusion as well as several other parameters such as the strength of the relationship between the statements, whether the premise is an assumption, an exception, or a simple premise. Complex graphs can be constructed to represent a range of arguments and their interconnections. The graphic can then be exported as an XML file.

4.1.7 Application Program Interfaces

An application programming interface (API) is a set of declarations of the functions (or procedures) that is provided to enable computer programs to make requests for support and services from an operating system, library, or web service. The Estrella API provides a standard set of functions to interact with Carneades allowing programs to assert facts and rules, to ask queries, to ask for explanations, and so on.

The reference inference engine of the Estrella project is Carneades, which is written in Scheme. In order to make the inference engine more generally accessible, an interface allowing access by clients using languages such as Java and C has been provided. The technology chosen is SOAP, a WC3 recommendation since 2003. SOAP transports XML encoded requests and responses over HTTP and uses the web services description language WSDL to provide well defined descriptions of the services requested. Essentially a web service wrapper for Carneades is provided by linking the scheme implementation to a gSOAP server application written in C through a C/Scheme language interface, also written in C.

This enables clients to access the services provided by the Carneades inference engine over the internet using the established SOAP protocol.

4.1.8 The Inference Engine

The Open Source inference engine developed in the Estrella project is intended to serve as a model illustrating one way to generate arguments from concepts, rules, and cases modeled in LKIF.

The reference inference engine can generate arguments about:

- the relationships among legal concepts.
- make inferences about the properties of individuals (e.g. what is implied in a given legal system where an individual is a pensioner).
- reason about rules which may change or which have exceptions (i.e. are *defeasible*) or are prioritised.
- reason about cases.

The inference engine cannot only generate arguments from LKIF knowledge bases, but also help users evaluate the acceptability of claims, taking into consideration the given arguments, the allocation of the burden of proof, and the applicable proof standards.

The engine also provides tools for visualising arguments. The engine uses resource-bound heuristic strategies to search the space of arguments. The code is written in a declarative, functional style using the Scheme programming language.

4.1.9 Document Management System

Legal documents constitute a central source of legal knowledge. To represent legal documents in the EU, it is required to provide a jurisdiction-independent XML standard for interchange and automated processing. A central requirement of automated systems based on representations of legal knowledge is an explicit link to the original legal sources of the knowledge (traceability). The link should maintain a consistent relationship between the knowledge base, arguments about legal decisions, and the legal sources. Moreover, the link supports efficient maintenance of the knowledge base as the law changes. It should allow one to validate that the legal knowledge representation system does what it claim; it should also support justifications of decisions [Boer et al., 2008].

Representing the legal sources and linking them to a knowledge base is not a trivial task. Legal sources come in many forms and formats, from different jurisdictions using different legal systems, in different languages, with different internal structures, etc. Legal sources are not self-contained entities but are related to each other; that is, laws are in temporal relations (laws which precede other laws), in jurisdictional relations (one law in a jurisdiction depending on another law in another jurisdiction), in hierarchical relations (the powers of a legal hierarch within a juridiction), and others. Moreover, there is little consensus about how to represent legal knowledge in different jurisdictions.

To address these issues a Document Management System has been provided using both MetaLex XML and LKIF. In MetaLex, legal sources are stored in a single, uniform, and general Content Management System architecture. It is a lowest common denominator for other standards which imposes a standardized view on the data which thereby supports software development. It allows that rules that are specific to a jurisdiction can be captured with addition elements of the architecture. Legal documents are then portable and yet localisable. For example, it can represent legal documents using existing mature national XML based services for valid legislation like the Norme-in-Rete service in Italy or AWB in the Netherlands, and yet translate to other XML format languages in other jurisdictions.[1] Finally, the system is extensible so as the requirements of the legal system change, the management system can change as well.

4.2 The Setting in Artificial Intelligence and the Law

To this point, we have reviewed the key design elements of the Estrella Project. In this section, we show that the elements are firmly based on research which has emerged from work on artificial intelligence and law. In this section we will give a brief pedigree of the various strands of the Estrella Project, locating them in the AI and Law tradition. For our research sources we will focus on work presented at the series of International Conferences on AI and Law (ICAIL). Held every other year since 1987, these conferences have provided an excellent series of snapshots of emerging research in AI and Law.

In the 1987 conference a major theme was developing practical, expert systems style applications based on logical representations of legislation as discussed in [Bench-Capon et al., 1987]. This work was pursued actively by a number of groups, and by the 1991 conference there was a good understanding of the kinds of application amenable to this approach, typically areas which required complex legislation to be applied to a large volume of cases, where the difficulty resulted from aware-

[1]Norme-in-Rete: http://www.normeinrete.it
AWB: http://wetten.overheid.nl

ness of the legislation and the interaction of different pieces of legislation rather than nuances of interpretation, such as welfare benefits and tax law. By 1991 the focus was on how to build such applications, and stressed on the importance of relating code to the legal sources, using so-called isomorphic [Bench-Capon, 1991] or verbatim [Johnson and Mead, 1991] representations. This understanding was then applied commercially, aware that its success was more dependant on the development of methodology and tools for constructing and deploying application than in technological refinement. This strand of work appears in Estrella through Haley, who have long been engaged in the commercial development of tools and techniques to support this approach. Further research on techniques for modelling legislation, based on the same engineering principles was carried out in the POWER project [van Engers and Boekenoogen, 2003], and this work is represented in Estrella by Rulewise.

Also a long standing topic is the visualisation of legislation and legal arguments, and one approach to this is the use of decision trees, which can be traced back to [Greenleaf et al., 1987]. This approach forms the basis of the Knowledge Tools system. Other visual representations, such as that in the ZENO system [Gordon and Karacapilidis, 1997], the argument diagramming tool developed in Arucaria [Reed and Rowe, 2004], and Carneades [Gordon et al., 2007a], are reflected in the argument graphs produced by the Estrella inference engine.

With rule based systems now being developed in a commercial context, research effort shifted to attempts to model and exploit more subtle aspects of legal reasoning. At this time in general AI there was a trend towards the development of ontologies [Gruber, 1993], and in AI and Law this gave rise to the view of legal problem solving as mapping from the concepts of a real world ontology into the concepts of a legal ontology (e.g. [Breuker and den Haan, 1991]). Since then developments in tools for building ontologies, a greater understanding of ontologies, such as the role of core and top ontologies, and, very importantly, description logic provers such as Pellet for reasoning with ontologies, have made this once theoretical approach a practical proposition, and it forms the basis for a strand of work in Estrella.

The other key approach to legal reasoning in AI and Law in the middle 90s was the use of argumentation to handle the defeasibility of legal conclusions (e.g. [Prakken, 1993]). This topic has received much attention in handling the logic of defeasibility, in using argumentation schemes to allow more natural expression of argument, and in enriching the types of arguments that can be put forward. Argument schemes were introduced using the scheme of Toulmin ([Lutomski, 1989], [Marshall, 1989], [Bench-Capon and Staniford, 1995]), but have been extended greatly over the years. The inference engine of Estrella is firmly grounded on this research into argumentation and uses a generalised version of argument schemes as in [Gordon et al., 2007b].

Also relevant is an awareness that different legal sources need to be reasoned about using different techniques. An early example of such a hybrid reasoner in AI and Law is [Oskamp et al., 1989]. In Estrella the inference engine is designed to reason with a variety of sources, including cases and ontologies. Mention should be made of cases in particular: there is a long history of exploration of arguing with cases in AI and Law, including HYPO [Rissland and Ashley, 1987] and CATO

[Aleven and Ashley, 1995]. In Estrella the approach of these systems has been modelled to integrate case based reasoning with the Estrella inference engine.

To summarise: there has been a long tradition of research in AI and Law, and the Estrella Project has drawn together a number of important developments in this field. The latest manifestations of a range of these approaches taken from AI and law research have been developed into the tools delivered on this project to provide an integrated environment.

Chapter 5

Translators

In this chapter, the industrial and academic partners briefly describe in their own way and terms their rule languages, explain what parts of LKIF are targeted by each translator, and outline what LKIF does for them.[1]

5.1 Haley, UK

Haley Office Rules (formerly RuleBurst Studio) is a business rules management system that combines propositional logic with a natural language parser to run web-based interviews with human users. Haley Office Rules gives users the ability to export their projects as a PIF (Project Interchange File). A PIF is a stand-alone XML file that represents the project attributes, business rules, and rule-associated data. Apart from the rules, much of the data in this file is associated with natural language parsing and is not directly pertinent to the inference engine. Conceptually, each rule is composed of exactly one 'conclusion' statement which is the root node of a tree of 'premise' statements. Each premise can be composed of more premise statements. The breadth and depth of this tree can be as large as necessary. In practice, the PIF XML is not structured like a tree but contains explicit 'level' attributes that indicate the depth at which the statement is located. This is the main structural difference between the two formats as LKIF theory rules are structured as trees from the start.

A LKIF - PIF translator was built to prove the interchange-ability of the two formats. Translating from a PIF to an LKIF is useful to Haley because it strips the PIF format of its Haley-specific project and natural language parsing data while still providing an expressive language that can represent Haley rules. The data lost in this translation is not pertinent to the representation of rules and can be re-derived after being imported back into Haley Office Rules. Even though there is a distinct difference between the intended uses of the two formats (propositional logic vs. predicate logic), LKIF is still able to properly express the Haley rules.

Because of its expressiveness however, not all of LKIF's features can be directly translated into a PIF. For instance, Haley Office Rules does not allow exception rules (rule:excluded). Likewise other features that do not have a direct translation can still be worked around.

[1] These descriptions have been lightly edited for style for inclusion in this report.

The Estrella project has provided Haley with the opportunity to collaborate with fellow consortium members on the formation of these new standards and how they might be applied in a real-world scenario. This knowledge keeps Haley at the forefront of emerging methodologies in the field. Haley is interested in adopting LKIF as a supported standard to allow customers freedom with regards to interoperability. Haley also has many European Government clients interested in open standards for rules such as:

- Inside Europe: Forsakringskassan, Danish SKAT, HMRC, BusinessLink, Swedish Tax, Dutch Statistics, Swedish Social Care
- Outside Europe: US IRS, Singapore MinDef, Australian Tax
- Other sectors: BUPA, Nuon, BAT, and others

Haley is keen to be at the forefront of technology. In addition, it is important to Haley to win and keep customers not by restricting choice but by supporting them in being the best in their field. Haley's involvement in the Estrella project will certainly help it towards this goal.

5.2 University of Budapest, Hungary

AllexGold is the representation language into and out of which LKIF was translated. The language is built on the concept of *frames*, which is similar to the object-oriented paradigm of classes and objects. Each frame represents an *object* of the modeled domain: it has a set of so-called *attributes* which describe its properties. Attributes can be of several types. While frames describe the actors of the domain, their relations given by *rules*. A rule consists of a set of conditions about attribute values and a conclusion which is an assertion about the value of another attribute. The values/properties attributes or the frames can be queried, and mathematical formulae about attribute values is supported.

The main differences between AllexGold and LKIF are:

- AllexGold uses frames, i.e. every model implements a kind of hierarchy among its domain objects and gives an explicit description of them. Such a systematic description of knowledge can be of great utility. This feature is missing from the core of LKIF.
- LKIF supports the use of predicates of arbitrary arity, a feature missing from AllexGold. Therefore, it cannot define inter-frame relations.
- LKIF supports defeasible rules, which is supported only partially by AllexGold through its default values of attributes.
- LKIF supports the inference of rule priorities while AllexGold only provides static priorities.

Those aspects of the languages which they have in common are translated:

- LKIF domain constants can be represented by AllexGold frames.
- Unary predicates in LKIF can described by the attributes of AllexGold frames.
- Strict LKIF rules (i.e. non-defeasible) can directly be transformed to and from AllexGold rules. Because of differences between the two kinds of rules, it might be required to represent one LKIF rule by multiple AllexGold rules or vice versa.

5.3 knowledgeTools

knowledgeTools is a logic visualisation and inference system where basic statements that are either true or false are represented in an *AND*/*OR* logic tree with the usual semantics. There are two sorts interpretation of *OR*: with *inclusive OR* , the inference follows from the truth of *any* premise or possibly all; with *exclusive OR*, the inference follows from the truth of *only one* premise. Topmost statements can be investigated by traversing down the logic tree (backwards chaining).

In translating between the logic tree of knowledgeTools and LKIF, the aim was to retain the logical consequences of each representation with respect to the translatable elements. In exporting a tree into the LKIF format, the logical tree structure with respect to *AND* and *inclusive OR* were maintained along with the title of each node; expressions with *exclusive OR* are not translated. In importing a tree from the LKIF format, several transformations are required, for example, translating exceptions into negated premises. There are other particular differences between the formats.

The translation was tested against the taxation directive pilot which was represented in knowledgeTools then exported to LKIF. When reimporting, the resultant logic tree showed no loss of information. Furthermore, importing LKIF files provided by other pilot studies resulted in successful translations to knowledgeTools with some differences.

The representation of argument graphs in LKIF is flexible enough to describe knowledgeTrees. Converting back from general LKIF argument graphs to a form that matches knowledgeTrees is somewhat more challenging. These observations highlighted ways in which the translation mechanism could be improved in the future, for example the translation of arguments without premises.

5.4 University of Bologna – UNIBO

The work on the Italian Pilot was connected with the development, use, and validation of a translator to/from the LKIF language and the HALEY proprietary language. Both LKIF rules and arguments are supported by the HALEY translator, and in general it showed the capability to preserve the expressivity and logic of the source argument graphs or rules modelled in LKIF language.

Nevertheless, some issues appeared in the translation process from LKIF to Haley which were caused by the fact that the logic which Haley language is built upon is different from the logic of LKIF. In general terms, the logic implemented by LKIF has a more expressive syntax. However, the adoption of some guidelines in the representation of legal norms helped to completely overcome these issues. The following is a list of the issues faced and of the solutions adopted:

- LKIF allows rules with different premises to have the same conclusion. This is not allowed by Haley (multiply proven attribute error during the compilation). This has been solved by making all the LKIF rules with the same conclusion into a single Haley rule and using the "or" operator;

- LKIF allows a single rule to have multiple conclusions, which is not allowed by Haley. This has been solved by splitting the rule in a set of rules, each having one of the conclusions;

- LKIF allows the use of an *excluded* predicate in the conclusion of a rule, when this rule works as an exception of another rule. This predicate is not supported by Haley, but can be simulated by nesting the "exception rule" as a negative premise of the general rule.

5.5 RuleWise

RuleWise specializes in applying legal knowledge tools for rule management. To provide tailored services to its clients RuleWise not only makes use of its own methods and tools, but also utilizes other vendor's tools and methods.

The Rulewise method can be characterized as a knowledge acquisition and knowledge representation method. The resulting representation is used for testing the quality of legislation, for designing enforcement processes, and as the knowledge component within automated systems.

RuleWise employs the object-oriented representation language UML/OCL. Since this language has close links with what OWL provides, it was natural to translate into LKIF via OWL. The translation process from RuleWise to LKIF starts with an XMI export of a model from the RuleWise workbench. XMI is an XML for UML. An XSLT provides the mapping between this XML source file and the desired destination format, which is OWL. The actual translation of the RuleWise model in XMI to OWL occurs when running the XSLT. This can be done easily with standard tools which run XSLT's. For this pilot the MetalexStyler was used (provided on the Metalex web site). The translation from LKIF back to RuleWise uses the same XSLT approach. The translators were tested with good results in the European Directive pilot.

Participation in Estrella has provided RuleWise with several benefits. The translation into OWL brings the developments within the semantic web community within reach, such as the use of reasoners, querying languages, and case tools. Through LKIF, RuleWise can also apply the functionality of other vendors participating in Estrella, without having to (completely) remodel the content. Finally the collaboration in Estrella has provided RuleWise with insight in several legal knowledge tools, both commercial and open source. This all has broadened the palette of services RuleWise can provide to its clients.

RuleWise and her mother company O&I are determined to keep on contributing to the development of standardized languages and tools for legal knowledge support. As of the end of the Estrella Project, O&I participates in projects where the Estrella results will be exploited, together with, for example, the Immigration and Naturalization Office and the Dutch Tax and Customs Administration.

Chapter 6

Pilots

To demonstrate and validate the ESTRELLA platform, legislation from two European countries as well as EU directives were modeled and used for pilot applications across several industrial partners. The objective of the exercise was to determine the effect of a standardised framework on how the partners model domains, the degree to which LKIF increases efficiency and lowers costs, identify issues in designing translators, and determine what sort of information is lost in translation (if any). A two pronged strategy was deployed. The industrial partners modeled the target domain with their own tools and methods. The partners in academia and public administration also modeled the same domain directly in LKIF. This highlighted the expressivity, completeness, and correctness of the various models. In the following sections, participating partners report in their own way and terms on their experience modeling the target domain.[1]

6.1 University of Budapest, Hungary

To construct the Hungarian pilot model, a piece of legislation was selected, rules were extracted, and the rules were expressed in LKIF.

The focus of the model was legislation on the responsibilities and liabilities on paying value added tax (VAT) for transactions across EU member state boundaries was chosen. This was chosen because: (1) the other partners are also working on tax legislation; (2) the legislation is relatively small and easily separated from other parts of legislation, making it suitable for the role of an experimental domain; and (3) it involves relatively diverse kinds of knowledge processing, e.g. logical inference with possible exceptions as well as numeric calculations.

A set of informal rules were extracted from the legal source texts with the aid of an expert. These were rules formulated in natural language; however, they closely resembled their future LKIF versions. The central concept of the modeled domain is a transaction between a vendor and a buyer across EU member state boundaries. The goal of inference is to decide whether the vendor is liable for paying value added tax for the transaction in question. Assuming that the answer is 'yes', further issues can be examined. A decision is to be inferred from information such as: (1) legal properties of the vendor and the buyer, mainly concerning their taxation status,

[1]These descriptions have been lightly edited for style for inclusion in this report.

(2) properties and value of the sold goods, (3) method of transportation, and (4) summed value of further transactions by the vendor over the year.

LKIF rules were constructed from these informal rules. The set of domain actors were determined along with a set of elemental relations describing their connections. The properties of the actors were defined through the introduction of predicates to the domain. LKIF rules were constructed to describe the logical relations and connections between different properties and quantities in the domain. Finally the knowledge about the case at hands was described by facts of the knowledge base. After the knowledge base was constructed, it was tested for validity and usability using the reference inference engine *Carneades*.

In conclusion, the LKIF language is especially well-suited for the definition of the rules of legislation; it forms a natural means for such modeling tasks. Numeric calculations can also be formulated, however, with a bit of less ease. LKIF also fits well to the needs and expectations of human users, i.e. it can be regarded as a natural means for the encoding of legal knowledge.

6.2 knowledgeTools

knowledgeTools modeled the rights and the duties of the Member States in respect with the deferral of taxation of the capital pursuant to the Council Directive 2005/19/EC of 17th February 2005. In this directive, the addressee of the directive is a member state. The main goal is to determine whether the state is obligated to grant a preliminary deferral of the taxation of the capital gains in a case where a company is restructuring or transferring offices. A variety of additional goals were identified concerning the maintenance of tax exemptions and taxation on losses.

Given the analysis of the directive, an *AND/OR* logic tree was constructed. The main benefit from the logic tree is that it is easier to understand the law and to reason to conclusions using a tree than reading a text; one can focus on a portion of the tree and use it to reason systematically.

6.3 University of Bologna – UNIBO

The Italian Pilot centered on the representation of norms relating to taxation of savings income in the form of interest payments, both at the level of the original EU Directive (Council Directive 2003/48/EC) and of the Italian national law (Legislative decree No 48 of 18 April 2005).

The aim of the savings directive and Italian law is to determine when and how a "paying agent" is bound to collect some information regarding an "interest payment" made in favour of a "beneficial owner". The paying agent must also determine the minimum amount of information to be collected and then transmitted to a "competent authority". The directive and the national law have slightly different perspectives: the directive directed to member states as recipients of the information, while the national law is directed to all the entities and activities subject to the Italian law. This difference resulted in a different issues the representation of the normative content of the laws.

Since the representations were aimed at confronting of EU and Italian norms in a legal drafting perspective, a high degree of granularity was requested. They were developed to be isomorphic to the source documents to the maximum possible extent.

The work was divided in two tasks: (i) analysis of source norms and the identification of concepts, definitions, key actors, and their roles; (ii) knowledge representation, including testing. For both the EU directive and the Italian law, a portion of the text was selected for analysis.

The modelling was made using both the IDE provided with the Dr-Scheme system distribution and a commercial XML editor (XMLSpy). The rulebases were tested through the Carneades reference inference engine implemented in PLT-Scheme with sets of cases provided by experts from the Italian Ministry of Finance. Given several starting facts and relative final goals, the reference inference engine was able to generate answers consistent with the source norms.

In general the results of the work were satisfying, except for a discrepancy referable to a difference between the two laws caused by the choice of the Italian legislator to shift a norm from one article to another, thus modifying the logical connection of norms as expressed in the source EU Directive. The LKIF rule language, tested against a rather complex set of norms, proved to correctly capture the normative content of the source legislation.

6.4 RuleWise

RuleWise has modeled the Council Directive 2005/19/EC of 17th February 2005 about mergers and transfers. The RuleWise method was followed in modelling the directive. The method supports a compartmentalized way of working going from legal text to small models of one section and finally combining these into one domain model. This model has been translated to LKIF with the RuleWise translator. The RuleWise model of the Directive is an UML/OCL model which represents the Directive graphically and also provides links from model elements back to the original source text. Furthermore, discussions about the right interpretation of the legislation is kept in notes in the model.

Although the legislation was very complex, the compartmentalized way of working helped a great deal in reducing the complexity for the knowledge analyst. The graphical representation provided the analyst with the necessary overview of the model. For a citizen or a business, the main benefits of utilizing models like RuleWise models proved to be:

- More accessible legislation. The graphical representation combined with query gives the user a better overview of the legislation and also the means to extract exactly the desired information.

- Greater maintainability of knowledge models and systems in case of legislation change. Having a traceable model at their disposal means that users can better assess the impact of a change in legislation on the knowledge models. If this traceability is extended towards products and processes based on that legislation, this also makes it more straightforward to implement a change in the user's organisation induced by a change in legislation. It gives an actor the necessary support to conduct real portfolio management.

- A check on the quality of legislation. While modeling, all kind of mistakes in the legislation are found because the modeler makes the legislation clear and insightful. Running the model afterwards can also contribute to finding flaws in the legislation. If these mistakes are found at an early stage in the legislation chain, they can be corrected and further cost (e.g. through appeals) can be prevented.

- Building up the organizations' knowledge corpus on how to interpret the legislation. Ideally when modeling, the authorized domain experts in organizations are participating in the process. They use their expertise to reach decisions about the favoured interpretation of the legislation. Keeping track of these discussions and decisions can be very worthwhile for an organization because the conclusions can be re-used over time and by other persons.

Bibliography

[Aleven and Ashley, 1995] Aleven, V. and Ashley, K. D. (1995). Doing things with factors. In *ICAIL*, pages 31–41.

[Bench-Capon, 1991] Bench-Capon, T. J. M. (1991). Exploiting isomorphism: Development of a kbs to support british coal insurance claims. In *ICAIL*, pages 62–68.

[Bench-Capon et al., 1987] Bench-Capon, T. J. M., Robinson, G. O., Routen, T., and Sergot, M. J. (1987). Logic programming for large scale applications in law: A formalisation of supplementary benefit legislation. In *ICAIL*, pages 190–198.

[Bench-Capon and Staniford, 1995] Bench-Capon, T. J. M. and Staniford, G. (1995). Plaid: Proactive legal assistance. In *ICAIL*, pages 81–88.

[Boer et al., 2008] Boer, A., Winkels, R., and Vitali, F. (2008). Metalex xml and the legal knowledge interchange format. In Sartor, G., Casanovas, P., Rubino, R., and Casellas, N., editors, *Computable Models of the Law: Languages, Dialogues, Games, Ontologies*, LNAI 4884, page To Appear. Springer.

[Breuker and den Haan, 1991] Breuker, J. and den Haan, N. (1991). Separating world and regulation knowledge: Where is the logic. In *ICAIL*, pages 92–97.

[Estrella Project, 2007] Estrella Project (2007). Specification of the legal knowledge interchange format. Technical report, University of Amsterdam.

[Gordon, 2007a] Gordon, T. (2007a). Visualising carneades argument graphs. *Law, Probability, and Risk*, 6(1-4):109–117.

[Gordon et al., 2007a] Gordon, T., Prakken, H., and Walton, D. (2007a). The carneades model of argument and burden of proof. *Artificial Intelligence*, 171:875–896.

[Gordon, 2007b] Gordon, T. F. (2007b). Constructing arguments with a computational model of an argumentation scheme for legal rules: interpreting legal rules as reasoning policies. In *ICAIL*, pages 117–121. ACM.

[Gordon, 2008a] Gordon, T. F. (2008a). Constructing legal arguments with rules in the legal knowledge interchange format. In Sartor, G., Casanovas, P., Rubino, R., and Casellas, N., editors, *Computable Models of the Law: Languages, Dialogues, Games, Ontologies*, LNAI 4884, page To Appear. Springer.

[Gordon, 2008b] Gordon, T. F. (2008b). Hybrid reasoning with argumentation schemes. presented at Computational Models of Natural Argument.

[Gordon and Karacapilidis, 1997] Gordon, T. F. and Karacapilidis, N. I. (1997). The zeno argumentation framework. In *ICAIL*, pages 10–18.

[Gordon et al., 2007b] Gordon, T. F., Prakken, H., and Walton, D. (2007b). The carneades model of argument and burden of proof. volume 171, pages 875–896.

[Greenleaf et al., 1987] Greenleaf, G., Mowbray, A., and Tyree, A. (1987). Expert systems in law: The datalex project. In *ICAIL*, pages 9–17.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220.

[Johnson and Mead, 1991] Johnson, P. and Mead, D. (1991). Legislative knowledge base systems for public administration: Some practical issues. In *ICAIL*, pages 108–117.

[Kirschner et al., 2003] Kirschner, P., Buckingham Shum, S., and Carr, C., editors (2003). *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making. London: Springer Verlag.* Springer.

[Lutomski, 1989] Lutomski, L. S. (1989). The design of an attorney's statistical consultant. In *ICAIL*, pages 224–233.

[Marshall, 1989] Marshall, C. C. (1989). Representing the structure of a legal argument. In *ICAIL*, pages 121–127.

[Oskamp et al., 1989] Oskamp, A., Walker, R. F., Schrickx, J. A., and van den Berg, P. H. (1989). Prolexs divide and rule: A legal application. In *ICAIL*, pages 54–62.

[Prakken, 1993] Prakken, H. (1993). A logical framework for modelling legal argument. In *ICAIL*, pages 1–9.

[Reed and Rowe, 2004] Reed, C. and Rowe, G. (2004). Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, 13(4):961–980.

[Rissland and Ashley, 1987] Rissland, E. L. and Ashley, K. D. (1987). A case-based system for trade secrets law. In *ICAIL*, pages 60–66.

[van Engers and Boekenoogen, 2003] van Engers, T. M. and Boekenoogen, M. R. (2003). Improving legal quality - an application report. In *ICAIL*, pages 284–292.

[Wigmore, 1931] Wigmore, J. H. (1931). *The Principles of Judicial Proof.* Little, Brown and Company.

[Wyner, 2007] Wyner, A. (2007). An ontology in OWL for legal case-based reasoning. In Atkinson, K., editor, *Proceedings of the JURIX 2007 Workshop on Modelling Legal Cases*, Leiden, The Netherlands.

[Wyner and Bench-Capon, 2007] Wyner, A. and Bench-Capon, T. (2007). Argument schemes for legal case-based reasoning. In Lodder, A. R. and Mommers, L., editors, *Legal Knowledge and Information Systems. JURIX 2007*, pages 139–149, Amsterdam. IOS Press.