Applications and Innovations in Intelligent Systems XIII

Ann Macintosh, Richard Ellis and
Tony Allen (Eds)

Applications and Innovations in Intelligent Systems XIII

Proceedings of AI-2005, the Twenty-fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, December 2005

Springer

Professor Ann Macintosh, BSc, CEng
Napier University, Edinburgh, EH10 5DT, UK

Richard Ellis, BSc, MSc
Stratum Management Ltd, UK

Dr Tony Allen, PhD
Nottingham Trent University, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN-10: 1-84628-223-3 Printed on acid-free paper
ISBN-13: 978-1-84628-223-2

Printed in the United Kingdom

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springer.com

APPLICATION PROGRAMME CHAIR'S INTRODUCTION

A. L. MACINTOSH
Napier University, UK

The papers in this volume are the refereed application papers presented at AI-2005, the Twenty-fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, held in Cambridge in December 2005. The conference was organised by SGAI, the British Computer Society Specialist Group on Artificial Intelligence.

This volume contains fifteen refereed papers which present the innovative application of a range of AI techniques in a number of subject domains. This year, the papers are divided into sections on *Applied AI in Information Processing*, *Techniques for Applied AI*, *Industrial Applications* and *Medical Applications*.

This year's prize for the best refereed application paper, which is being sponsored by the Department of Trade and Industry, was won by a paper entitled "Case-Based Reasoning Investigation of Therapy Inefficacy". The authors are Rainer Schmidt of the University of Rostock, Germany, and Olga Vorobieva, from the Sechenow Institute in St.Petersburg, Russia.

This is the thirteenth volume in the *Applications and Innovations* series. The Technical Stream papers are published as a companion volume under the title *Research and Development in Intelligent Systems XXII.*

On behalf of the conference organising committee I should like to thank all those who contributed to the organisation of this year's application programme, in particular the programme committee members, the executive programme committee and our administrator Collette Jackson.

Ann Macintosh
Application Programme Chair, AI-2005

ACKNOWLEDGEMENTS

AI-2005 CONFERENCE COMMITTEE

Dr Tony Allen, Nottingham Trent University	(Conference Chair)
Dr Alun Preece, University of Aberdeen	(Deputy Conference Chair, Electronic Services)
Dr Nirmalie Wiratunga, The Robert Gordon University	(Poster Session Organiser)
Professor Adrian Hopgood, Nottingham Trent University	(Workshop Organiser)
Professor Ann Macintosh, Napier University	(Application Programme Chair)
Richard Ellis, Stratum Management Ltd	(Deputy Application Programme Chair)
Professor Max Bramer, University of Portsmouth	(Technical Programme Chair)
Dr Frans Coenen, University of Liverpool	(Deputy Conference Chair, Local Arrangements and Deputy Technical Programme Chair)
Rosemary Gilligan	(Research Student Liaison)

APPLICATION EXECUTIVE PROGRAMME COMMITTEE

Professor Ann Macintosh, Napier University (Chair)
Richard Ellis, Stratum Management Ltd (Vice-Chair)
Rosemary Gilligan, University of Hertfordshire
Professor Adrian Hopgood, Nottingham Trent University
Mr Richard Wheeler, University of Edinburgh

APPLICATION PROGRAMME COMMITTEE

CONTENTS

APPLICATION KEYNOTE ADDRESS

Legal Engineering: A structural approach to Improving Legal Quality[1]

Tom M. van Engers[2]
University of Amsterdam/Leibniz Center for Law
vanEngers@uva.nl
www.LeibnizCenter.org

Abstract

Knowledge engineers have been working in the legal domain since the rise of their discipline in the mid-eighties of the last century. Traditionally their main focus was capturing and distributing knowledge by means of the knowledge-based systems, thus improving legal access. More and more legal knowledge engineering has become an analytical approach that helps to improve legal quality. An example is the POWER-approach developed in a research programme that is now finished. This programme was run by the Dutch Tax and Customs Administration (DTCA in Dutch: Belastingdienst) and some partners (see e.g. Van Engers et al., 1999, 2000, 2001, 2003 and 2004). The POWER-approach helped to improve quality of (new) legislation and codify the knowledge used in the translation processes in which legislation and regulations are transformed into procedures, computer programs and other designs. We experienced that despite these clear benefits implementation proved to be far from easy. In fact the implementation phase still continues. Adapting research results in public administrations is a tedious process that takes lots and lots of energy and requires continuous management attention. Learning at organisational level proved to be much harder than we thought.

1. Introduction

Getting the right knowledge at the right place at the right time has always been a great challenge for governments since this inflicts the ability to effectuate the legislative power to regulate and control. The Dutch Tax and Customs Administration (DTCA) has developed a method and supporting tools supporting the whole chain of processes from legislation drafting to executing the law by government employees and citizens (see e.g. Van Engers et al., 1999, 2000, 2001,

[1] Parts of the material described here have been published before (see e.g. Van Engers 2004).

[2] Tom M. van Engers was programme manager of the POWER research programme and coordinator of the IST funded E-Power project.

2003 and 2004). These method and tools resulted from the POWER research program (Program for an Ontology-based Working Environment for Rules and regulations), a research program that was partly sponsored (the E-POWER project) by the European Commission through the IST 5th framework program.

The POWER approach combines two typical knowledge management approaches; the stock or codification approach and the flow or organisational approach. It offers both a method and supporting tools that support a systematic translation of (new) legislation into the administrations' processes.

The motive behind running the POWER program was that drafting and implementing new legislation is a rather time, energy and money consuming process consisting of many inter-connected processes. These processes are very vulnerable to errors. Not only because of the intrinsic complexity of the law, but also because mostly a large number of people is involved in these processes as wells as of the complexity of these processes themselves. Varying interests have to be aligned and communication difficulties due to differences in technical jargon have to be overcome in both drafting and implementing changes to legislation. The same holds when completely new legislation has to be implemented.

The POWER-method helps to improve the quality of (new) legislation. It also supports codification of the legal knowledge into procedures, computer programs and other designs. One of the advantages thereof is the reduction of the time-to-market of the implementation of legislation and its increased transparency which will lead to reduced maintenance costs and improved services to citizens.

The POWER-approach was developed in a research programme that is now finished. In this paper I will share some of my experiences with you without explaining the POWER-approach in too much detail.

2. Design of regulatory processes

The knowledge and experience needed to create new laws or adapt existing ones, specify, design and implement procedures and systems in legislative domains is very scarce. A (piece of) law should reflect the intentions of the political responsible minister and should also meet some quality criteria such as clarity and consistency from the perspective of the law-enforcement organization. This is the responsibility of the legislation drafters that are responsible for drafting the new law.

The people responsible for implementing the law (i.e. the administration) have to adapt the procedures, processes and information systems to the new law. Also risk diagnosis, assessment procedures and audit measurements have to be designed and implemented as well. This all has to be done while taking political and social-environmental requirements into account. An example of such requirements is the need for diminishing the administrative costs for citizens and businesses.

Between drafting new legislation and enforcement thereof a chain of processes has to be managed and aligned. Preventing errors as early as possible in this chain can save a lot of time and money. Not only at the design stage but even more during the law-enforcement stage. Unintended use or even worse abuse is often due to anomalies in the law. Also, the position of the government is much stronger when involved in a dispute if the law is very clear with respect to the object of disagreement.

Many legislation drafting departments at the different ministries already have their own quality insurance techniques. Furthermore in many cases the ministry of Justice has a special role because they are usually responsible for the overall legal quality of a country. Despite all the effort that's been spend on improving legal quality using traditional measurements, such as co-reading (peer reviewing etc.) many anomalies can still be found in recently drafted legislation. The situation is even worse in situations when existing legislation is adapted.

Quality insurance measurements also exist for the other processes in the chain mentioned. Most attempts to achieve quality improvements however focus at just one of the processes involved. In the approach developed in the POWER-research program we consider each of these processes as equally important. We furthermore stress the importance of managing the chain rather then the distinctive processes themselves.

Finding a way to improve legal quality was just one of the three main goals of the POWER research program. The other two goals are reduction of total cost of ownership (TTO) of the (knowledge-based) systems intended for the support of civil servants or citizens and secondly, reduction of time to market (TTM) i.e. the speed with which these (knowledge-based) systems can be created, and consequently the regulatory power can be effectuated.

The POWER-approach supports the finding of anomalies in legal sources. Central in the approach is the central role for formal (and semi formal) knowledge representations. In the POWER-approach different knowledge representation formats are used. How these knowledge representations are used and how they contribute to improving legal quality is extensively described in other papers (see e.g. Van Engers et al., 1999, 2000, 2001, 2003 and 2004). Amongst them are both procedural descriptions called 'scenarios' (which are more or less comparable to UML action diagrams) and POWER-conceptual models (expressed in UML/OCL). Although scenarios (see section 3) lack the benefits of a strict formal model expressed in UML/ OCL (van Engers et al., 2001 [3] and [4]) they are useful to provide both analysts and experts with a good insight in the legal domain represented, especially when the legislation involved is to be used in a categorization or assessment task. Scenarios also proved to be an excellent means of communication with experts and representatives of disciplines involved in the implementation of legislation (see Van Engers et al. 2002 [7]).

The POWER approach combines two typical knowledge management approaches; the stock or codification approach and the flow or organisational approach. It offers both a method and supporting tools that support a systematic translation of (new) legislation into the administrations' processes. POWER offers a systematic approach can help to improve legal quality.

In contrast to other knowledge modelling approaches the POWER-approach is focused on modelling legal sources rather than expert knowledge. Expert knowledge however is needed to find the correct interpretations but also for efficiency reasons. Starting with representing the (legal) experts' knowledge (using scenarios) helps to find the adequate scope (the legal sources to be analysed). Confronting the expert with differences between the model build out of the experts' knowledge and the knowledge that can be distilled out of the other knowledge sources (specifically the law) causes the legal experts to see things in a different light and has often led to changes in the law.

3. Quality improvement

The quality of the law enforcement depends on the quality of the legislation itself and on the quality of the knowledge systems that are actually used in the client handling processes as well. In previous work (see Van Engers and Boekenoogen 2003) we described some results from a project that was aimed a improving the quality of legislation and the investigation of the consequences of implementing a new law. Others also have proposed approaches form quality improvement both in the legal domain as for knowledge in general (see e.g. Voermans 2000, Preece 1994, Vanthienen 1997 and Spreeuwenberg 2001).

The problem with most verification procedures is that these procedures can only be applied after most of the hard work has already been done, i.e. formalising the knowledge. Since this formalisation process, i.e translating the legislation into a formal representation (van Engers et al., 2001 [5]) and applying a verification process to it, usually takes some time even when it is supported by tools such as the POWER-workbench, a less subtle and profound approach is needed to satisfy the practical needs of legislation drafters and policy makers need feedback. Especially if in the drafting process, where these drafters deal with the politicians and other influential stakeholders feedback is needed in a much earlier stage. Furthermore it is not always necessary to design a (rule-based) system at all. Specifically for this purpose we developed a less labour and time intensive method derived from the original Power-method that helps to find anomalies. That method is called the Power-light method. We can choose to just applying the Power-light if we don't need a formal model for the purposes mentioned before.

The POWER-approach is used to detect anomalies in legislation, but as a regular part of the design of the regulatory processes not as a separate step. Under time pressure we can use the Power-light method which works quite similar, except for the fact that formalization only takes place in the modellers' mind.

How the POWER-method can be used to detects in legislation defects has already been describe in previously published papers (e.g. Van Engers 2004) I refer to those papers for more detailed information about the process and examples. Typical anomalies found are circularities, ambiguous references and missing concepts, gaps in the law and inapplicable regulations.

4. Knowledge representation

Central to the POWER-approach is the creation of a formal knowledge representation of the legal domain at hand. This representation is called the POWER-conceptual model. The representation formalism, design and examples are already described in previous publications (e.g. Van Engers et al 2001 [5]) for readers yet unfamiliar with this approach in this section a very short introduction is given. The POWER-conceptual model is represented in a notation called Unified Modelling Language (UML see D'Souza and Wills 1999). This notation has become one of the most accepted standard notations for representing domain model, but there are many ways to use the notation. The usage defined in the POWER-method, starts by dividing the model in UML packages. The structure of packages within the translated conceptual model is identical to the hierarchy in the legislation (i.e. chapters, sections, articles, members etc.), which allows tracing all conceptual models, and products that will derive from them, to the original legislation. The structure of packages within the integrated conceptual model represents the definition of concepts found in the legislation, and the relationships between these definitions.

Within each UML package, the important concepts found in the legislation are modelled as types and attributes. The references found in the legislation are modelled as an extension to the UML, which we called "Package Reference". A package reference is modelled as a classifier, which represents some not-yet-identified other packages. Finally, the norms within the legislation are modelled in a formal language, named Object Constraint Language (OCL), which is a part of the UML. The Object Constraint Language can for instance determine under which conditions a "Natural Person" becomes a "Tax Payer". One can use OCL in a similar way as one would use a reified first order predicate calculus to express a legal norm.

The conceptual models produced this way (the POWER-model) contain the legal knowledge. When this knowledge is combined with the process and task knowledge, we have a specification for a supporting knowledge-based component.

Before making a formal representation of a certain domain it proved to be helpful to first understand a bit about the legal domain. This is best obtained by looking at how some (prototypical) cases are solved that correspond to a certain target group. The reasoning strategies of legal experts used for the solution of these (hypothetical or real) cases can be represented in a kind of procedural representation like a decision tree which can be expressed in e.g. UML action diagrams. Within the POWER research program we use a special form of such action diagrams which we call 'scenarios' because they represent the possible scenarios of solving cases (see also Van Engers et al. 2002 [7]).

We experienced that developing such scenarios at the start of the knowledge modelling process helped both knowledge analysts and legal experts to understand the domain better, especially in case of modelling new or complex legislation.

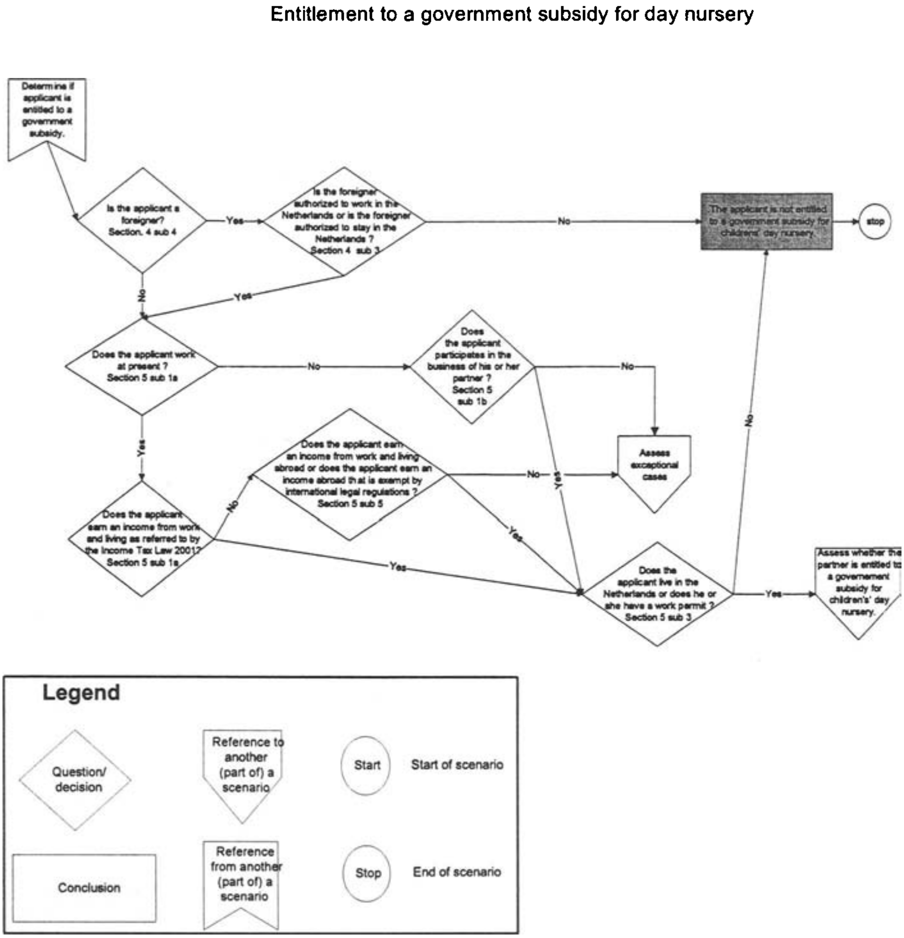

Figure 1. Example of a (part of) a scenario called "determine entitlement to a government subsidy for day nursery"

Many approaches exist in which these very decision diagrams *are* the main knowledge representation (and interface from the modelling tool to the knowledge engineer). Examples of such tools are amongst others, Ruleburst, Knowledge Tools and Be-Informed Studio.

POWER scenarios provide a pure functional idea of how legal experts use legislation to solve certain cases: it does not provide a "system view" on how a decision support system would function. This is the distinction between a POWER scenario and a UML scenario, which is a story about how a system will be used. A UML scenario describes a prototypical sequence of interactions in a business-collaboration or the system context (D'Souza et al., 1999). The main difference between POWER scenarios and UML scenarios is that UML scenarios are used to define the boundary of a system, whereas POWER scenarios can be considered as a (global) specification of the knowledge intensive process (which could be supported by a system). However different, UML scenarios and POWER scenarios match when it comes to the goal of capturing the task flow. UML scenarios are used to capture the ideal task flow as perceived by end users. POWER scenarios provide us with the means for discovering the implicit tasks and task flow within legal domains.

The idea of using these scenarios as the basis for knowledge-based systems design may be tempting, but essential to the POWER approach is that we base our knowledge models on the legal sources rather than on the experts' interpretation of these sources. As we found out the experts' interpretations may be incomplete or even conflict with these knowledge sources (i.e. in conflict with what is the law!). Furthermore a serious handicap of procedural representations is their limitations they put on the implementation. The order of the different reasoning steps represented in the scenarios may very well be not the most efficient one. Also when designing a user dialogue for a knowledge-based system one may want to choose a different order for posing questions then you would derive form such scenarios.

5. Conclusions

The POWER-method has shown to be a very useful approach for modelling normative systems. These systems are described in laws and other regulations including regulations that are used outside the government e.g. insurance policies. The POWER-method offers a structural design method for improving regulatory processes. One of its strong advantages is the support for detecting anomalies in legislation in an early stage of design (preferably even before the law becomes effective). This makes the Power-method a powerful tool for both legislation drafters, law-enforcement organizations and other organizations that are responsible for the design and/or execution of large bodies of regulations.

We experienced that despite these clear benefits implementation proved to be far from easy. In fact the implementation phase still continues. The transfer from knowledge from research projects such as the POWER programme into the regular processes of huge organisations as public administrations cost lots of energy and is time consuming. Changing people's way of working, demands a change of mind-set that not everyone can cope with. Another problem also not specific to the POWER-approach is the lack of managers with an information science back-ground in public administrations. Usually the managers are recruited with a completely different skill set. Therefore these managers simply don't understand the relationships between the features of the design processes (including methods, tools, staff and infrastructure) and the quality of the public administrations primary processes. Implementation of approaches such as the POWER-approach should therefore also be considered as an organisational change programme, demanding all kind of actions including educating the managers. I learned that learning at an individual level can be hard sometimes, but learning at corporate level is much harder.

References

1. Van Engers, T.M., Kordelaar, P., 1999, POWER: Programme for an Ontology based Working Environment for modeling and use of Regulations and legislation. Proceedings of the ISMICK'99.
2. Van Engers, T.M., Kordelaar, P.J.M., den Hartog, J., Glassée, E., 2000, POWER: Programme for an Ontology based Working Environment for modeling and use of Regulations and legislation. In Proceeding of the 11th workshop on Database and Expert Systems Applications (IEEE) Eds. Tjoa, Wagner and Al-Zobaidie, Greenwich London, ISBN 0-7695-0680-1, pp. 327-334.
3. Van Engers, T.M., Gerrits, G., Boekenoogen, M.R., Glassée E., Kordelaar, P.,2001, *POWER: Using UML/OCL for Modeling Legislation – an application report*. Proceedings of the ICAIL 2001, St. Louis.
4. Van Engers, T.M., Glassée, E., 2001, Facilitating the Legislation Process using a shared Conceptual Model. In: IEEE Intelligent Systems January 2001, p. 50- 57.
5. Van Engers, T.M., Van Driel, L., Boekenoogen, M., 2002, The effect of formal representations formats on the quality of legal decision-making, in Legal Knowledge and Information Systems (Bench-Capon, Daskalopulu, Winkels eds.), IOS Press Amsterdam, ISBN 1 58603 299 2, p 63-71.
6. Van Engers, T.M., Boekenoogen, M.R., 2003, Improving Legal Quality: an application report, in proceeding of ICAIL2003.
7. Van Engers, T.M., Legal engineering: A knowledge engineering approach to improving legal quality. In J. Padget, R. Neira, and J.L. De León, editors, *eGovernment and eDemocracy: Progress and Challenges*, pages 189-206. Instituto Politéchnico Nacional Centro de Investigacion en Computación, 2004. ISBN 970-36-0152-9.
8. Glassée, E., Van Engers, T.M., Jacobs, A., 2003 POWER: An Integrated Method for Legislation and Regulations from their Design to their Use in E-government Services and Law Enforcement, in Digital Legislation (ed Moens), die Keure, Brugge, ISBN 90 5958 039 7, p175-204
9. Preece, Shingal, 1994, Foundation and Application of Knowledge Base Verification, International Journal of Intelligent Systems, 9, 683 – 701
10. D'Souza D.F. and Wills A.C., 1999, Objects, components and frameworks with UML: the Catalysis approach, Addison-Wesley, ISBN 0-201-31012-0.
11. Spreeuwenberg, S., Van Engers, T.M., Gerrits, R.,2001, The Role of Verification in Improving the Quality of Legal Decision-Making, in Legal Knowledge and Information Systems, IOS press, ISSN 0922-6389.
12. Vanthienen, J., Mues C., Wets, G., 1997, Inter-Tabular Verification in an Interactive Environment, Proceedings Eurovav 97, 155 – 165
13. Voermans, W., 2000, Computer-assisted legislative drafting in the Netherlands: the LEDA-system, A National Conference on Legislative Drafting in the Global Village

BEST APPLICATION PAPER

Case-Based Reasoning Investigation of Therapy Inefficacy

Rainer Schmidt[1], Olga Vorobieva[1+2]

[1] Institute for Medical Informatics and Biometry, University of Rostock, D-18055 Rostock, Germany
[2] Sechenov Institute of Evolutionary Physiology and Biochemistry, St.Petersburg, Russia

Abstract. In this paper, we present ISOR, a Case-Based Reasoning system for long-term therapy support in the endocrine domain and in psychiatry. ISOR performs typical therapeutic tasks, such as computing initial therapies, initial dose recommendations, and dose updates. Apart from these tasks ISOR deals especially with situations where therapies become ineffective. Causes for inefficacy have to be found and better therapy recommendations should be computed. In addition to the typical Case-Based Reasoning knowledge, namely former already solved cases, ISOR uses further knowledge forms, especially medical histories of query patients themselves and prototypical cases (prototypes). Furthermore, the knowledge base consists of therapies, conflicts, instructions etc. So, retrieval does not only provide former similar cases but different forms and steps of retrieval are performed, while adaptation occurs as an interactive dialog with the user. Since therapy inefficacy can be caused by various circumstances, we propose searching for former similar cases to get ideas about probable reasons that subsequently should be carefully investigated. We show that ISOR is able to successfully support such investigations.

1 Introduction

In medical practice, therapies prescribed according to a certain diagnosis sometimes do not give desired results. Sometimes therapies are effective for some time but suddenly stop helping any more. There are many different reasons. A diagnosis might be erroneous, the state of a patient might have changed completely or the state might have changed just slightly but with important implications for an existing therapy. Furthermore, a patient might have caught an additional disease, some other complication might have occurred, or a patient might have changed his/her lifestyle (e.g. started a diet) etc.

For long-term therapy support in the endocrine domain and in psychiatry, we have developed a Case-Based Reasoning system, named ISOR, that not only performs typical therapeutic tasks but also especially deals with situations where therapies become ineffective. Therefore, it first attempts to find causes for inefficacy and subsequently computes new therapy recommendations that should perform better than those administered before.

ISOR is a medical Case-Based Reasoning system that deals with the following tasks:

- choose appropriate (initial) therapies,
- compute doses for chosen therapies,
- update dose recommendations according to laboratory test results,
- establish new doses of prescribed medicine according to changes in a patient's medical status or lifestyle,
- find out probable reasons why administered therapies are not as efficient as they should,
- test obtained reasons for inefficacy and make sure that they are the real cause, and
- suggest recommendations to avoid inefficacy of prescribed therapies.

ISOR deals with long-term diseases, e.g. psychiatric diseases, and with diseases even lasting for a lifetime, e.g. endocrine malfunctions.

For psychiatric diseases some Case-Based Reasoning systems have been developed, which deal with specific diseases or problems, e.g. with Alzheimer's disease [1] or with eating disorders [2]. Since we do not want to discuss various psychiatric problems but intend to illustrate ISOR by understandable examples, in this paper we focus mainly on some endocrine and psychiatric disorders, namely on hypothyroidism and depressive symptoms. Inefficacy of pharmacological therapy for depression is a widely known problem (e.g. [3, 4, 5, 6, 7]). There are many approaches to solve this problem. Guidelines and algorithms have been created (e.g. [8, 9, 10]). ISOR gives reference to a psychopharmacology algorithm [10] that is available on the website htp://mhc.com/Algorithms/Depression.

The paper is organized as follows. Firstly, we introduce typical therapeutic tasks, subsequently we present the architecture of ISOR and finally we illustrate its work by examples.

2. Methods: Typical Therapeutic Tasks

As a consequence of our experiences with ICONS [11], a system for antibiotic therapy advice, and with therapy support programs for hypothyroidism [12], we believe that four tasks exist for medicinal therapies. The first one means computing an initial therapy, secondly an initial dose has to be determined, later on dose updates may be necessary, and finally interactions with further diseases, complications, and especially with already administered therapies have to be considered.

In the following we illustrate the four tasks by our programs that deal with therapy support for hypothyroid patients. The antibiotics therapy adviser ICONS deals only with two of these tasks: computing initial therapies and initial doses.

2.1 Computing an initial therapy

Probably, the most important task for therapies is the computation of initial therapies. The main task of ICONS is to compute promising antibiotic therapies even before the pathogen that caused the infection is determined in the laboratory. However, for hypothyroidism ISOR does not compute initial therapies but only initial doses, because for hypothyroidism only one therapy is available: it is thyroid hormone, usually in form of levothyroxine.

2.2 Computing an initial dose

In ICONS the determination of initial doses is a rather simple task. For every antibiotic a specific calculation function is available and has to be applied.

For hypothyroidism the determination of initial doses (figure 1) is more complicated. Firstly, a couple of prototypes exist. These are recommendations that have been defined by expert commissions [13]. Though we are not sure whether they are officially accepted, we call them guidelines. The assignment of a patient to a fitting guideline is obvious because of the way the guidelines have been defined. With the help of these guidelines a range for good doses can be calculated.

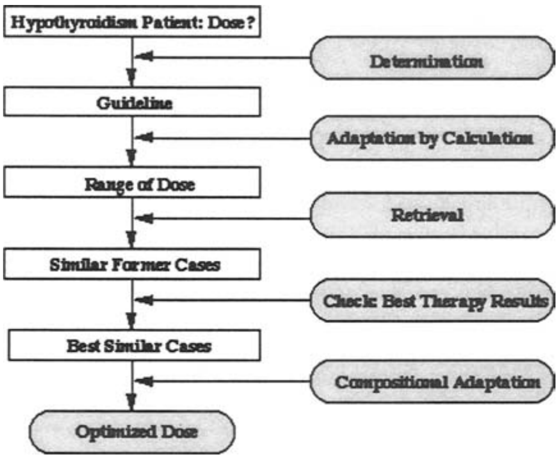

Fig. 1. Determination of an initial levothyroxine dose

To compute a dose with best expected impact, we retrieve similar cases whose initial doses are within the calculated ranges. Since cases are described by few attributes and since our case base is rather small, we use Tversky's sequential measure of dissimilarity [14]. On the basis of those retrieved cases that had best therapy results an average initial therapy is calculated. Best therapy results can be determined by values of a blood test after two weeks of treatment with the initial dose. The opposite idea to consider cases with bad therapy results does not work here, because bad results can also be caused by various other reasons.

To compute optimal dose recommendations, we apply two forms of adaptation. First, a calculation of ranges according to guidelines and patients attribute values. Secondly, we use compositional adaptation. That means, we take only similar cases with best therapy results into account and calculate the average dose for these cases, which has to be adapted to the query patient by another calculation.

2.3 Updating the dose in a patient's lifetime

For monitoring a hypothyroidism patient, three basic laboratory blood tests (TSH, FT3, FT4) have to be undertaken. Usually the results of these tests correspond to each other. Otherwise, it indicates a more complicated thyroid condition and additional tests are necessary. If the results of the basic tests show that the patients thyroid hormone level is normal, it means that the current levothyroxine dose is OK. If the tests indicate that the thyroid hormone level is too low, the current dose has to be increased by 25 or 50 μg, if it is high, the dose has to be decreased by 25 or 50 μg [15, 16]. So, for monitoring, adaptation means calculating according to some rules, which are based on guidelines. Since an overdose of levothyroxine may cause serious complications for a patient, a doctor cannot simply consider test results and symptoms that indicate a dose increase but additionally he/she has to investigate reasons why the current dose is not appropriate any more. In ISOR this situation is described as a problem of therapy inefficiency. In most cases the solution is obvious, e.g. puberty, pregnancy etc. These situations are covered by adaptation rules. Sometimes cases are observed in which the hypothyroidism syndromes are unexplained. For these cases ISOR uses the problem solving program.

2.4 Additional diseases or complications

It often occurs that patients do not only have hypothyroidism, but they suffer from further chronic diseases or complications. Thus, a levothyroxine therapy has to be checked for contraindications, adverse effects and interactions with additionally existing therapies. Since no alternative is available to replace levothyroxine, if necessary additionally existing therapies have to be modified, substituted, or compensated (figure 2) [15, 16].

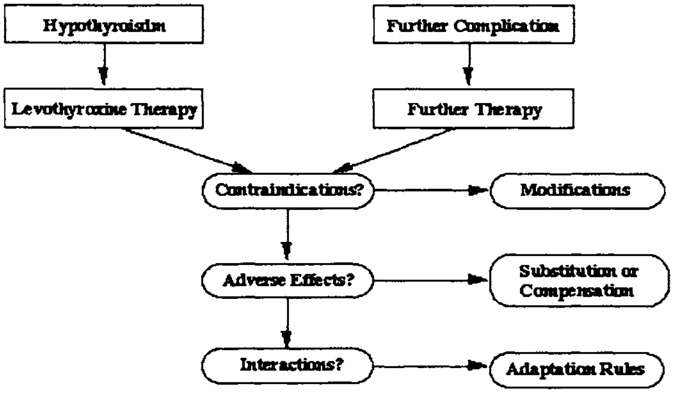

Fig. 2. Levothyroxine therapy and additionally existing therapies

ISOR performs three tests. The first one checks if another existing therapy is contraindicated to hypothyroidism. This holds only for very few therapies, namely for specific diets like soybean infant formula, which is the most popular food for babies who do not get enough mother's milk but it prevents the effect of levothyroxine. Such diets have to be modified. Since no exact knowledge is available to explain how to accomplish this, our program just issues a warning saying that a modification is necessary.

The second test considers adverse effects. There are two ways to deal with them. A further existing therapy has either to be substituted or it has to be compensated by another drug. Such knowledge is available, and we have implemented corresponding rules for substitutional and compensational adaptation.

The third test checks for interactions between both therapies. We have implemented some adaptation rules, which mainly attempt to avoid the interactions. For example, if a patient has heartburn problems that are treated with an antacid, a rule for this situation states that levothyroxine should be administered at least four hours after or before an antacid. However, if no adaptation rule can solve such an interaction problem, the same substitution rules as for adverse effects are applied.

3 System architecture

ISOR is designed to solve typical problems, especially inefficacy of prescribed therapies that can arise in different medical domains. Therefore most algorithms and functions are domain independent. Another goal is to cope with situations where important patient data is missing and/or where theoretical domain knowledge is controversial.

ISOR does not generate solutions itself. Its task is to help users by providing all available information and to support them to find optimal solutions. Users shall be doctors, maybe together with a patient.

Technically, ISOR is implemented in Delphi 7, the format for the case and knowledge bases is Paradox 7, and retrieval is performed by SQL.

In addition to the typical Case-Based Reasoning knowledge, namely former already solved cases, ISOR uses further knowledge components, namely medical histories of query patients themselves and prototypical cases (prototypes). Furthermore, ISOR's knowledge base consists of therapies, conflicts, instructions etc. The architecture is shown in figure 3.

In this section we explain the components and in the next chapter we present examples to show how the main knowledge components work together.

3.1 Medical case histories

Ma and Knight [17] have introduced a concept of case history in Case-Based Reasoning. Such an approach is very useful when we deal with chronic patients, because often the same complications occur again, former successful solutions can be helpful again, while former unsuccessful solutions should be avoided.

The case history is written in the patient's individual base as a sequence of records. A patient's base contains his/her whole medical history, all medical information that is available: diseases, complications, therapies, circumstances of his/her life etc. Each record describes an episode in a patient's medical history. Episodes often characterise a specific problem. Since the case base is problem oriented, it contains just episodes and the same patient can be mentioned in the case base a few times, even concerning different problems.

Information from the patient's individual base can be useful for a current situation, because for patients with chronic diseases very similar problems often occur again. If a similar situation is found in the patient's history, it is up to the user to decide whether to start retrieval in the general case base or not.

In endocrinology, case histories are designed according to a standard scheme, one record per visit. Every record contains the results of laboratory tests and of an interrogatory about symptoms, complaints and physiological conditions of a patient.