The integration of retrieval, reasoning and drafting for refugee law: a third generation legal knowledge based system

John Yearwood
School of Information Technology and Mathematical Sciences
University of Ballarat, Ballarat, Australia
+61 353 279 270
j.yearwood@ballarat.edu.au

Andrew Stranieri
School of Information Technology and Mathematical Sciences
University of Ballarat, Ballarat, Australia
+61 353 279 440
a.stranieri@ballarat.edu.au

ABSTRACT

We identify an argument to be the basic unit of reasoning of a system that supports the construction of arguments and drafting of determinations in refugee law. Collaboration with the Refugee Review Tribunal of Australia has led to the development of a framework for argument construction that includes over 200 generic arguments. However, these arguments may not encompass all arguments used in any particular case. The construction of non-generic arguments involves the integration of information retrieval within reasoning. This retrieval is passage based from a wide variety of text sources. The framework also acts as the illocutionary structure in a document drafting process. In conceptualising this system we have found it useful to propose a classification of knowledge based systems in law.

Keywords

Argument structure, reasoning, information retrieval, document drafting.

1. INTRODUCTION

Research in the application of IT to Law has traditionally been directed independently towards modelling legal reasoning, document retrieval, document drafting and legal intelligent tutoring. Research in each of these fields initially led to the development of systems which we call first generation systems. Systems which we call second generation systems have attempted to overcome the deficiencies of the first generation systems but have not generally attempted to integrate these separate tasks. Systems which we call third generation systems are based on a framework that facilitates the integration of reasoning, retrieval, drafting and tutoring.

In this paper we present a classification of legal knowledge based information systems into four generations with a particular focus on a framework for a third generation system in the domain of refugee law. In refugee law there is a need to access information from a variety of different document sources, to reason with that information and ultimately to draft refugee determinations. In order to model the decision making process of Refugee decision makers it is necessary to address the integration of information retrieval with legal reasoning and document drafting.

The context in which our third generation system operates, is the support of decision makers in refugee law in constructing arguments. The argument based approach is different from the case based approach commonly used in law in the sense that there is a focus on a unit of reasoning. Not only does the framework allow the retrieval of symbolically similar arguments and similar textual argument fragments but also the generation of new arguments.

Three systems that we classify as third generation systems are PLAID by Bench-Capon *et al* [4], ARMOR by Matthijssen [11] and CATO by Aleven and Ashley [1, 2]. Each of these systems attempts some level of integration of information retrieval with reasoning and drafting. Each system does not represent a fixed line of reasoning but models argumentation. Each system responds to user direction. These systems differ in their application domains, the structure of underlying knowledge, human-computer interaction design and inferencing methods.

In this paper we describe EMBRACE currently under development as a third generation system. This system differs from the three mentioned above in four main ways:

- Information retrieval is tightly integrated within the context of the reasoning.
- The system assists a user in the construction of an argument for or against refugee status for an applicant.
- The system itself can infer an outcome for refugee status based on inferences drawn by neural, fuzzy, logic, statistical or rule based reasoning. Users may thus compare and contrast their arguments with those inferred by the system.
- The system can be extended into systems we call fourth generation system.

In the following section we describe the concept of 1st, 2nd, 3rd and 4th generation legal knowledge based systems before illustrating the refugee system we are developing.

ICAIL-99 Oslo, Norway Copyright (c) 1999 ACM 1-58113-165-8/99/6 $5.00

2. GENERATIONS

Broadly speaking the majority of research into LKBS has been focussed on the independent development of reasoning systems, information retrieval systems and document drafting systems. By using a generational conceptualisation we see that higher generation LKBS require the integration of these three applications with each other and with end users.

Rule based systems exemplify first generation reasoning systems. In these systems domain knowledge is represented in a symbolic way that is largely independent of the broader context of users tasks or abilities. Zeleznikow and Hunter [21] provide a survey of rule based systems. First generation document drafting systems are essentially template driven. Applications such as *Scrivener* [10] provide an example of the extent to which useful systems can be developed with the incorporation of rules that represent limited domain knowledge. First generation information retrieval systems are those in which queries are specified using keywords and boolean or positional qualifiers and relevant documents are retrieved on the basis of statistical matching of keywords with words in documents.

Common threads can be drawn between, first generation reasoning, information retrieval and document drafting systems. Generally, first generation systems ignore human computer interaction (HCI) issues. Furthermore, domain knowledge is represented in a passive and static way, if at all. This has led to the knowledge acquisition bottleneck in reasoning systems. Rose [16] describes limitations for IR that arise when human computer issues are ignored and knowledge about documents is not represented. Branting *et al* [5] indicate the importance of explicitly representing knowledge about the intended use and stylistic conventions for document drafting.

Second generation reasoning systems can be seen as those in which knowledge bases are not a static representation of the world. The case based reasoning system of Ashley [3] represented the first and arguably, still the most sophisticated advance from the first generation reliance on a static and passive knowledge base. In more recent years other advances in reasoning systems have similarly moved away from a static representation of knowledge to one that is more dynamic. This is evident in the preponderance of argumentation based models. Gordon [8] and Prakken and Sartor [15] are just two notable examples in this direction.

In integrating a symbolic representation of documents with a subsymbolic one in the Scalir system, Rose [16] has overcome limitations of the first generation IR systems. The IR improvements made by Daniels and Rissland [6], Moens *et al* [14] and Yearwood [19] attribute improvements over first generation systems to the integration of more sophisticated representation of knowledge of documents with statistical techniques.

In a similar vein, the document drafting approach of Branting *et al* [5] moves away from the first generation systems by incorporating a rich representation of the purpose of the document for a user and of stylistic conventions that encompass rhetorical and thematic conventions. The document drafting system developed by Daskalopulu and Sergot [7] extends on first generation approaches by introducing an extensive representation of the domain.

All second generation systems aim to support one restricted aspect of a users tasks. For example, the rule/neural hybrid of Zeleznikow and Stranieri [20] makes a prediction in family law. Lawyers, judges, para-legal decision makers, mediators and litigants are all end users that find the development of an accurate prediction of a court outcome useful. However, in practice, a lawyer, for instance makes a prediction but also drafts documents and searches for information in order to provide a service to their clients. The lawyer does not predict with an internal reasoning system, search for precedents with an internal IR system or draft a document with an internal drafting system. Instead a lawyer applies family law knowledge in an integrated way. The knowledge for reasoning informs an information search and drafting and vice-versa.

The development of information retrieval techniques from those focused on document surrogates (abstracts and summaries) to full texts and more recently to passages within documents could also be considered a third generational focus of these systems. This is consistent with the approach of viewing a document as a collection of arguments which are passages.

Approaches that we label third generation systems are those that attempt to model the broader task in which the user is situated. Thus, knowledge is represented in such a way that the information retrieval is situated in the context of the reasoning for which the information is sought. This leads to improved information retrieval. Conversely, reasoning is improved by facilitating a search for information at the appropriate time. Drafting is improved by an integration with reasoning and information retrieval in a similar vein. Three examples of third generation systems are PLAID developed by Bench-Capon [4], ARMOR developed by Matthijsen [11] and the work with CATO by Aleven and Ashley [1, 2]. In this paper we present a different third generation approach that is currently under development in the domain of refugee or asylum seeker law.

Conceptualising LKBS developments in a generational framework lends itself to an articulation of our beliefs about the future direction of these systems. For example, we view fourth generation systems as those that add autonomy and reactivity to the integration of reasoning, information retrieval, document drafting and tutoring of third generation systems. Fourth generation systems will, in our view, be autonomous agents that initiate their own actions and, to some extent exercise their own judgement in order to support the work of other human and non-human agents. These systems will be able to interact in a variety of ways with a number of other agents thus facilitating the development of intelligent group decision support systems.

3. EMBRACE: AN INTEGRATION OF REASONING WITH RETRIEVAL AND DRAFTING

3.1 Refugee law

Each year, vast numbers of individuals lodge applications to remain in Australia for fear of persecution if forced to return to their country of origin. Their claims are assessed on the basis of the United Nations Convention Relating to the Status of Refugees

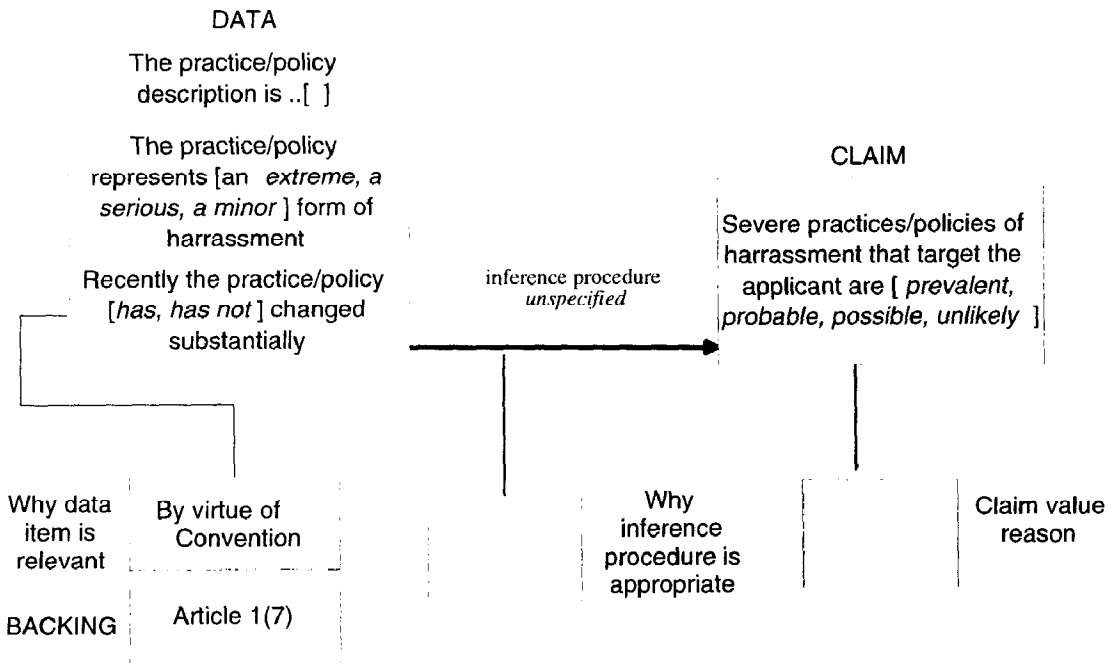

***Figure 1*: Toulmin structure variation for the Practices and Policies Argument**

(1951) by the Department of Immigration and Ethnic Affairs (DIEA), and on appeal, by the Refugee Review Tribunal (RRT).

Refugee law is highly discretionary and extremely difficult to model. The United Nations Convention lists factors that are to be taken into account in reaching a determination but does not specify the weighting factors should have. For example, the convention recognises that an applicant must have a well founded fear of persecution on grounds of political opinion, race, religious beliefs, social group membership, or nationality but is silent on the interpretation or the relative weighting of these factors. Each nation that is a signatory to the Convention has some case law that assists decision makers in interpreting key terms. For example, the test for *well founded fear* of persecution has been interpreted by the High Court of Australia as a real chance of persecution. The way key terms are interpreted depends on past RRT decisions, appellate Court decisions in addition to background information regarding the current political climate in an applicant's country of origin. Refugee law therefore, could be referred to as an open textured and discretionary domain. The text cases in refugee law discuss the main relevant features of the law and present the reasons for reaching the decision on refugee status.

3.2 Knowledge Representation for the Integration

Knowledge regarding refugee determination is represented using an argument based representation. We do not represent reasoning as a dialogue between applicant and member but instead model the reasoning that each agent, applicant, member and knowledge system, uses independently, to infer a final refugee outcome. That is, in our view, an applicant, in making a case for refugee status is presenting the Tribunal with an argument. A Tribunal member, in preparing a determination, is also making an argument for (or against) refugee status for that applicant. A knowledge based system employing machine learning methods is also making an argument for (or against) refugee status for the applicant.

An argument is represented using a frame that is based on, but not identical to, the structure proposed by Toulmin [18]. This frame has been used in earlier studies in modelling family law knowledge in Australia by Zeleznikow and Stranieri [20] and is illustrated in Figure 1.

Data elements in Figure 1 are the one or more elements such as *Recently, practices and policies in the country of origin,* that comprise the data component of the argument. Each data element has possible data element values. For example, the data element *Recent practices and policies in the country of origin* can take values on a 2 point scale, either "have not substantially changed" or "have changed substantially". The claim element in this argument is *Severe practices and policies of harassment that target the applicant* and the claim element values are either *"are prevalent"* or *"are possibly likely" or "are unlikely"*.

3.3 Components of EMBRACE

EMBRACE consists of 4 major modules: the knowledge base which we refer to as the generic argument structure, repositories an I/O module and an inferencing module. The knowledge base or generic structure is a framework of the arguments that are found in the domain. Part of this structure is shown in Figure 2. It is central to all components of EMBRACE. The repositories consist of a database of symbolically represented arguments, a database of text fragments corresponding to these arguments, a database of the full text of the cases that these arguments were extracted from and a repository of inference procedures. The symbolically represented arguments are linked to the text fragments which in

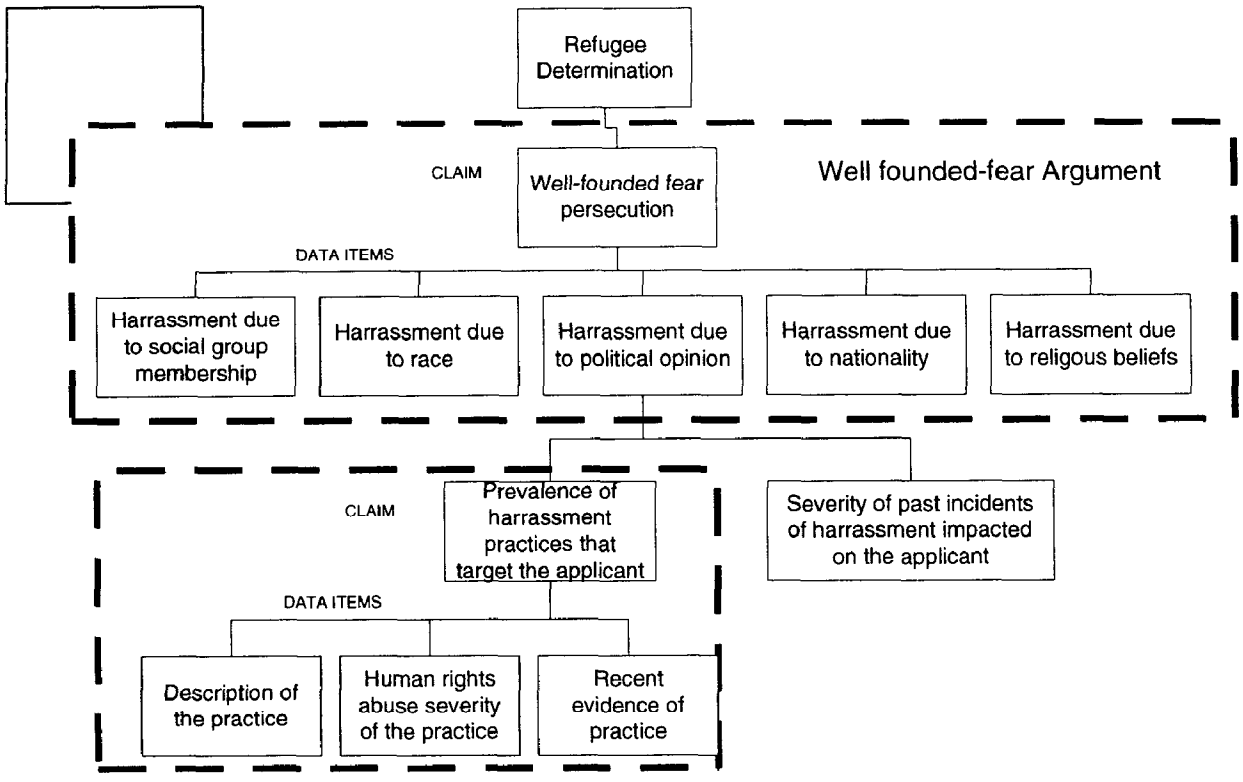

Figure 2: **Portion of argument structure for refugee law**

turn are linked to the full text documents. As well there is a repository of reports from organisations such as Amnesty International and Reuters. There is also a storage area for arguments under construction. The I/O module consists of two components: the claims notepad and the document drafter. The claims notepad is an interface to facilitate the representation and storage of the current argument as an instance of the generic argument structure. The document drafter facilitates the drafting of the determination by mapping the argument instance to a document along with the fragments of text that have been retrieved in the construction of the argument. The reasoning module invokes an appropriate inference procedure for each argument in the current argument so that the system can also provide the user with an outcome on refugee status.

Figure 3 depicts repositories for four types of knowledge used in EMBRACE. The generic argument base is a database containing those arguments known to be frequently used within the domain. Over two hundred of arguments like that depicted in Figure 1 have been identified after extensive consultation with RRT members as generic arguments. Each generic argument has a claim, data items, reasons for why each data item is relevant, the name of the associated inference procedure and reason for its appropriateness. A generic argument does not have a claim value reason because a value for the claim variable has not been set.

A trade off exists in the size of the generic argument repository. The open textured nature of legal reasoning mitigates against the representation of all arguments in a domain as generic arguments. However, consistency of determinations from one member to another is expected to be enhanced if there exist a larger number of generic arguments in use by all members. Too few generic arguments risks inconsistency across members and also requires that considerable duplication of effort in creating new arguments is likely.

Each generic argument, names an inference procedure which is a set of rules, neural network, statistical technique or any other procedure that can produce a claim value given data item values. These procedures are stored in the inference procedure repository. Separating the inference procedures from the rest of the argument component enables EMBRACE to be used as a support system where all inferences are made by humans. This is particularly important in many parts of refugee law as practised in Australia where, for various reasons, it is imperative that the machine does not infer any conclusions whatsoever.

Although the main function of EMBRACE will be to support argument construction we are also interested in building into it the ability to infer an arguments claim values from its data values and be able to propagate this to a determination. The inferencing module will access the inference procedures that are used to model the ways in which members move from sets of claim data values to claim values within each argument. A range of inference procedures will be used for different arguments. Although Figure 1 indicates inference procedures and reasons for them, these have not yet been determined.

Actual arguments made by applicants or members are each instances of a generic argument and stored in the actual argument

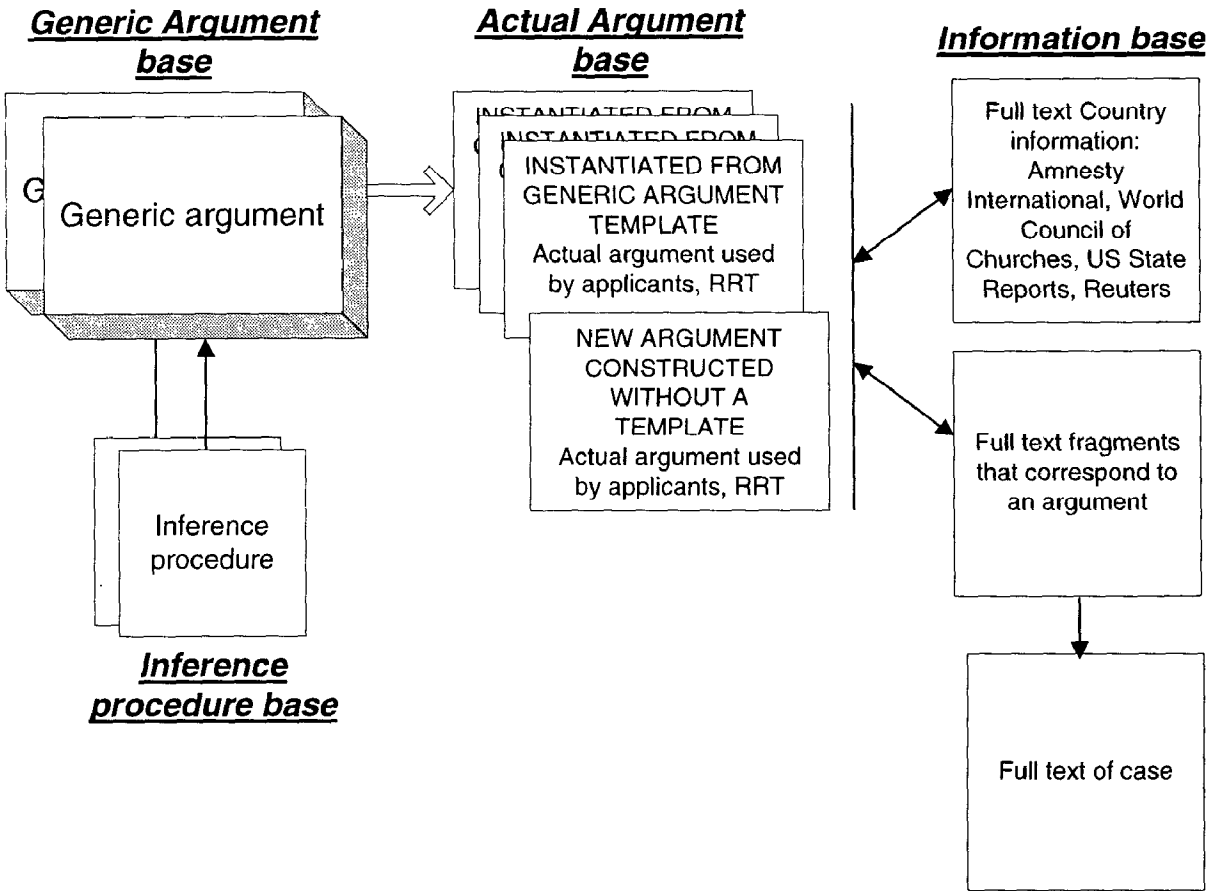

Figure 3: **Knowledge and information repositories in EMBRACE**

base. An instance of a generic argument is constructed by setting variable values and creating a new argument frame based on the generic template. Often an applicant asserts an argument for which no generic argument exists. In these cases, a new argument specific to that applicant is created. Ultimately, the series of actual arguments made by the applicant in support of refugee status is stored in the Actual argument base. Some of the arguments are instances of generic arguments, others are newly created. In creating a new argument that makes the required claim, a member first searches the actual argument base (symbolic and text) for an existing argument that has been made by another applicant or member in the past. If no such argument exists, the IR mechanism is invoked to search full text databases of country reports.

3.4 EMBRACE in action

EMBRACE is invoked in three distinct stages:

- A RRT member (or applicant for refugee status) uses EMBRACE to represent the chain of arguments an applicant has made in a written application for refugee status.
- A RRT member constructs a chain of arguments that supports or rebuts each argument in the chain proposed by the applicant
- A determination is drafted using the applicant's and member's argument chain.

The first phase involves creating an instance of the appropriate argument and entering data and claim element values according to the applicant's assertions. An instance of the Figure 1 argument where the applicant makes the claim that the LTTE Liberation Tigers of Tamil Elam) engage in systemic harassment of Tamils in Jaffna[1] is now considered. In support of this claim the applicant mentions that the LTTE regularly harass (non LTTE) Tamils in various ways. The applicant makes no mention of the severity of the harassment nor whether practices in Jaffna have recently changed which are the other two data items of the argument.

To support the member in representing this argument of the applicant, EMBRACE has a notepad interface where a member paraphrases the applicant's assertions. The notepad, currently under development, parses the assertion and attempts to identify the most appropriate argument in the structure. Once found, the member enters values for the appropriate data item and is reminded of adjacent data items. For example, once the member enters "LTTE engage in systemic harassment of Tamils in Jaffna" as the value for the first data item in the argument of Figure 1, he/she is prompted for any assertions regarding the severity of that harassment and whether the practice has changed recently.

[1] The applicant also makes many other claims of incidents of harassment that are captured in other parts of the argument structure.

Once an applicant's chain of arguments for refugee status is entered, the second stage is entered, where a member is prompted to agree, or disagree with each assertion made by the applicant. The member can pursue one of three courses of action for each argument; support the applicant's claim, attack the claim or explore further information. For example, a member may be unsure about the claim "*Severe practices and policies of harassment that target the applicant*" because they doubt the truth of the data item "the LTTE harasses Tamils in Jaffna". An argument that makes this latter claim needs to be found in existing cases or created. EMBRACE supports the member in this task in two ways :

- Assisting the member to find an existing argument by permitting an exact match of variable, value pairs in a symbolic database of arguments; for example the variable " Practice/policy description is.." and the value "the LTTE harasses Tamils in Jaffna"
- Assisting the member to create a new argument that has as its claim item "the LTTE harasses Tamils in Jaffna" by invoking an IR process that will search a full text database of past cases for fragments that correspond to arguments with this claim
- Assisting the member to create a new argument that has as its claim item "the LTTE harasses Tamils in Jaffna" by invoking an IR process that will search a full text database reports from organisations such as Amnesty International, Reuters and the World Council of Churches (WCC).

Figure 4 illustrates the argument structure that has the claim value "the LTTE harasses Tamils in Jaffna" that the member has created after invoking the IR engine on the database of reports. The reasons for this claim value are reports from Amnesty and WCC. The data items are extracted from the reports by the member.

The above example illustrates that the argument structure that has been developed not only describes the chains of reasoning that may be made but also facilitates the seeking of information to support or rebut arguments. The argument structure provides a framework so that:

- The retrieval is placed in context of the reasoning required
- The retrieval function can be contextualised
- The retrieval results provide contextual data for the reasoning
- The retrieval results may more precisely directed toward document drafting
- Rapid domain changes are catered for.

Currently most information seeking in this domain to support or refute the claims, claim values, data and data values is carried out by keyword search of a range of databases: previous cases, databases of country information, up-to-date news information (Amnesty, Reuters) and databases of high court cases. These databases are full text databases with some structure, although the structure is not one that explicitly parallels the argument structure used here.

The information seeking that is carried out in EMBRACE is argument based in the context of and to support reasoning. It is not whole case retrieval but a form of passage retrieval where the passages correspond to arguments within a case.

Yearwood [19] has shown that it is possible to automatically structure and index the database of previous cases to some level so that a more contextualised and effective retrieval function may be used. This requires some pre-processing of the text case-bases and case component indexing in addition to the full text indexing that would be carried out. There has been much work done on passage retrieval but mostly in the context of supporting whole document retrieval. Salton *et al*, [17] use the idea of requiring that documents first match a query in a global sense and then further that the greater the number of sub-section matches (paragraph or sentence) the stronger the evidence for retrieval of that passage. Mittendorf and Schäuble [13] consider document and passage retrieval based on Hidden Markov Models. Their passage retrieval model has some interesting parameters for controlling the passage length which is more effective in adjusting the right size of a passage than a rigid pre-segmentation into sentences. Kaszkiel and Zobel [9] approach the problem of document retrieval based on passages, systematically with the TREC Federal Register collection. They investigate arbitrary passage retrieval, where any sequence of words of any length starting at any word in the document is a valid arbitrary passage. The similarity of the highest ranked sequence of words from anywhere in the document (the arbitrary passage), being the document's similarity to the query. The results indicate that fixed-length arbitrary passages of 150 words or more perform significantly better than retrieval of whole documents with pivoted length normalisation of the cosine measure. The results also indicate that paragraphs, sections and tiles do not perform as well once pivoted length normalisation is used. No experiments on combining arbitrary passage level evidence with document level evidence were carried out.

SPIRE by Daniels and Rissland [6] is a hybrid case-based reasoning and information retrieval system which uses HYPO-style case templates for its initial retrieval stage and then uses the full text of these documents to retrieve other full text cases using INQUERY. Its second stage focuses on retrieving legally relevant passages from this small set of retrieved cases. This is carried out by focussing attention on overlapping text windows of sentence length (approximately 20 words). The construction of queries, by combining known relevant sections and expert descriptions, to retrieve the passages is quite similar to the method that we propose to use. However our indexing technique is based on both local and global indexing with a subsequent combination of evidence.

We propose the passage retrieval technique similar to that used by Yearwood in [19] and Daniels in [6], where a chunk of text based on a match with text that describes the current argument is used as a representative of that argument component in the case. The text used as a description of each of these arguments is taken from combinations of text from the LEGISLATIVE FRAMEWORK section of some cases, the convention and expert statements that capture the particular argument in our structure. The top matching paragraphs from each of the cases in the case-base is then used to form a text argument base for this argument component and this is used for queries relating to this argument component. This component level evidence based on local indexing (of the text within these components alone) will then be combined with a global match of the query with the full text of cases to produce the best retrieval of passages that will be tightly related to the argument component required. Some tuning of the technique is required as the size of the first level retrieved components has to

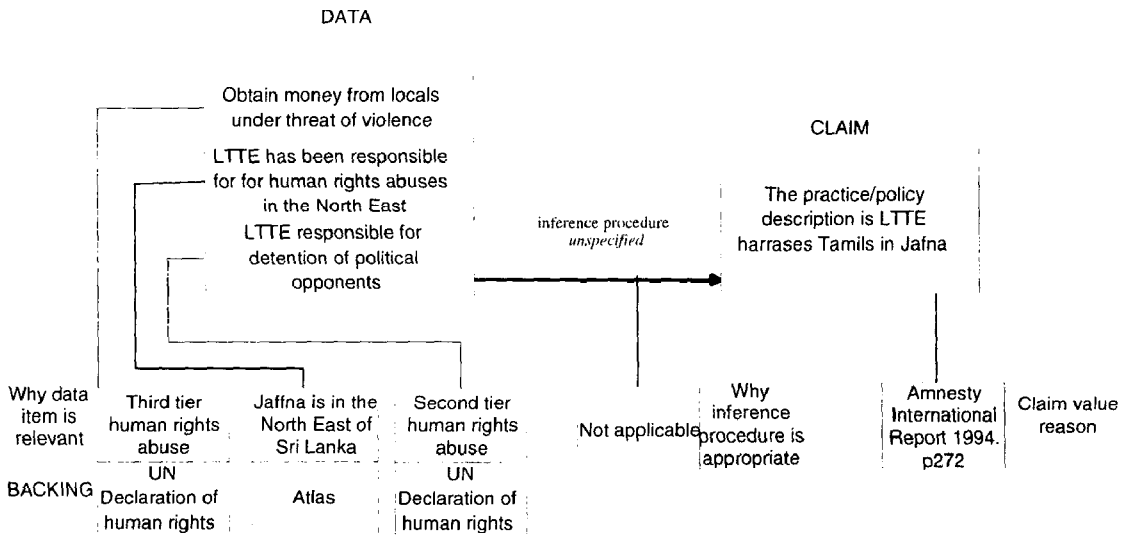

Figure 4: **The creation of a new argument**

be determined (and probably should be dynamic) and the optimum retrieval function also needs to be determined. Both of these tasks require learning based on a training set of cases.

The advantage is that the retrieval results can be more on point sections of cases as well as the full cases themselves. Not only do the on-point results more directly support reasoning but it is also likely that the text may be utilised in the document drafting process for the final written full text determination.

Based on the information obtained for each of the arguments made, the member builds a total argument of their own, which fits within the framework (or the need for new arguments is clearly indicated). The points of departure from the applicant's argument are easily identified within the framework. Once the information necessary to determine the individual data values and claim values has been collected it may be used by the system to automatically carry out the reasoning for each argument and hence a system determination can be generated. This may be used by members as a comparison and possibly provide indications as to whether their determination is in line with decisions of the past.

3.5 Drafting the Determination

Legal document drafting has sometimes been viewed as Computer Aided Design where the drafter uses blocks of text as the basic building blocks for the constructiom of the whole document. Some have taken the view that it is simply a form of document assembly (eg Scrivener [10]), others such as Daskalopulu [7] rely on an explicit representation of document structure with the creation of a new document instance being the process of assembling suitably instantiated blocks of text which comply with a set of explicit constraints.

Our argument structure or framework is a form of the illocutionary goal structure used by Branting *et al* [5] in the automated drafting of legal self explaining documents. This structure, as we have pointed out, facilitates the integration of information retrieval and reasoning by contextualising the information need and providing information to support the development of arguments. The claims notepad interface of EMBRACE has the main function of assisting the user to fill out the argument structure for the applicant as well as an argument structure for the members argument. However, the notes made, and the fragments of text retrieved which form the basis of various arguments can be stored and mapped to a document structure to form the basis of a draft determination. As we have seen in the example the data or support needed for an argument claim may be in the form of an appropriate piece of retrieved text. Frequently, this text may be used in the drafting process. The retrieved text or user modification of this can be drafted into a document which will form the basis for the final determination by the specification of an appropriate rhetorical structure. The illocutionary structure ensures that the goals and sub-goals in terms of the whole argument are met. The rhetorical structure specifies the discourse conventions that will be used to present the argument as a document which may entail the satisfaction of other goals. So, for example, the current document genre for the presentation of Refugee determinations contains a section entitled BACKGROUND which contains much factual information that would relate to leaf nodes in the illocutionary structure. At this stage in the development of EMBRACE no work has been done on the drafting component.

4. RELATED WORK AND CONCLUSIONS

In the PLAID system, Bench-Capon [4] represents knowledge as arguments using a structure based on that proposed by Toulmin [18]. The system generates a *brief* which is a discussion of issues that is prepared for a decision-maker to be quickly made aware of key issues. There are numerous sources of information including statutes, commentaries and personal information databases that

are drawn upon by computer agents within PLAID. These agents are triggered by a user's need to find information for use as a claim, data, backing, warrant or other component of a Toulmin argument. The argument framework assists the user to construct an argument (brief) and focuses the retrieval of information. In addition, the user interface is based on a dialogue game and ultimately a Rapporteur agent generates a document in English from the network of Toulmin like arguments.

Matthijsen [12] also uses a knowledge structure based on that proposed by Toulmin. Like Bench-Capon [4] he is also concerned with assisting a user in the retrieval of information from many and disparate sources. A model of tasks that users engage in is used to direct the construction of arguments and retrieval of information for the appropriate components of the Toulmin argument. Each argument has relevant documents directly associated with it. These are retrieved as needed during the construction of a new argument. However, in addition to these *static* links to documents, ARMOR enables dynamic links. Dynamic links use keywords in the text of an argument as index terms in full text databases to dynamically retrieve relevant information.

CATO uses the HYPO [3] case base, retrieval engine and adaptation mechanism but also incorporates additional features such as extended *argument moves* and a factor tree. In so doing, the CATO system readily integrates a reasoning system (HYPO) into a computer assisted learning system. A document drafting system that improves on first generation drafting systems can easily be imagined due to the versatility of the HYPO case based reasoning foundation.

The EMBRACE framework developed not only allows the capturing of arguments within Refugee law but facilitates the integration of information gathering and retrieval for the purpose of argument construction and drafting. The system does not attempt to automatically interpret information retrieved relating to any specific argument but facilitates the validating and substantiating of data, claims and their values by the user. Relevant text portions can be automatically tagged with their role in a draft document to explain the determination. A draft determination will finally be edited by the member of the tribunal with little more effort than performing the reasoning steps involved.

The system has been compared with other decision support systems in law and classified as a third generation system because of it ability to closely integrate information retrieval with the reasoning. Future research aims to optimise the retrieval mechanisms, explore the use of machine learning techniques within each argument and to implement the entire system using world wide web technology.

5. ACKNOWLEDGEMENTS

We would like to acknowledge the generous assistance of John Vrachnas from the Melbourne branch of the Refugee Review Tribunal of Australia and Domenico Calabrò from Victoria Legal Aid in developing the generic argument structure.

6. REFERENCES

[1] Aleven V., and Ashley K. D., 1994. An Instructional Environment for Practicing Argumentation Skills. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, AAAI'94, pages 485-492, Seattle, WA, July.

[2] Aleven V., and Ashley K. D., 1996. How Different is Different? Arguing about the significance of Similarities and Differences, In Smith I and Faultings B (eds), *Advances in Case-based Reasoning, Third European Workshop, EWCBR'96*, Springer, Berlin, pages 2-15.

[3] Ashley, K., 1990. *Modelling Legal Arguments. reasoning with cases and hypotheticals.* Cambridge, Mass. , MIT Press

[4] Bench-Capon, T. J. M. and Staniford, G. 1995. PLAID - Proactive Legal Assistance. Proceedings of *Fifth International Conference on Artificial Intelligence and Law: an International Journal* . May 21-24. Boston. ACM Press, USA. pp81-87.

[5] Branting, K., Lester, J. C., and Callaway, C. B. 1997. Automated Drafting of Self-Explaining Documents. *Proceedings of the Sixth International Conference on Artificial Intelligence and Law.* Melbourne. ACM Press. New York. pp72-81

[6] Daniels, J., and Rissland, E., 1997. Finding Legally Relevant Passages in Case Opinions. *Proceedings of the Sixth International Conference on Artificial Intelligence and Law.* Melbourne. ACM Press. New York. pp29-38

[7] Daskalopulu A., Sergot M., A constraint-Driven System for Contract Assembly, In *Proceedings of The Fifth International Conference on Artificial Intelligence and Law*, pages 62-70, May 21 - 24, University of Maryland, ACM Press, 1995.

[8] Gordon, T. F., 1997. The Zeno Argumentation Framework. *Proceedings of the Sixth International Conference on Artificial Intelligence and Law.* Melbourne. ACM Press. New York. pp10-19

[9] Kaszkiel M. and Zobel J., Pasage Retrieval Revisited, In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pages 178-185, August 27-31, Philadelphia, USA, 1997.

[10] Lauritsen M., 1992. Technology Report: Building Legal Practice Systems with todays commercial authoring tools. In Artificial Intelligence and Law, 1:1:87-102.

[11] Matthijssen, L., 1996. A task-based hyperindex for legal databases. (in) Kralingen van, R. W., Herik van den, H. J., Prins, E. J., Sergot, M., and Zeleznikow, J., 1996. *Legal Knowledge Based Systems. Foundations of Legal Knowledge Systems. Jurix96.* Tilburg University Press. The Netherlands. pp59-76

[12] Matthijssen, L., 1997. An intelligent interface for legal databases. *Proceedings of The Fifth International Conference on Artificial Intelligence and Law*, pages 71-80, May 21 - 24, University of Maryland, ACM Press, 1995.

[13] Mittendorf E. and Schäuble P., Document and Passage Retrieval Based on Hidden Markov Models. In *Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval*, pages 318-327, July 3-6, Dublin, Ireland, 1994.

[14] Moens, M., Uyttendaele, C., and Dumortier, J., 1997. Abstracting of Legal Cases: The SALOMON Experience. *Proceedings of the Sixth International Conference on Artificial Intelligence and Law.* Melbourne. ACM Press. New York. pp114-122

[15] Prakken, H., and Sartor, G., 1997. Reasoning with Precedents in a Dialogue Game. *Proceedings of the Sixth International Conference on Artificial Intelligence and Law.* Melbourne. ACM Press. New York. pp1-10.

[16] Rose, D. E, 1993. A Symbolic and Connectionist Approach to Legal Information Retrieval. Lawrence Erlbaum, Hillsdale, New Jersey.

[17] Salton G., Allan J., and Buckley C., Approaches to Passage Retrieval in Full Text Information Systems. In *Proceedings of the 16th Annual International Conference on Research and Development in Information Retrieval*, pages 49-58, June 27 - July 1, Pittsburg, USA, 1993.

[18] Toulmin, S. 1958. *The Uses of Arguments.* Cambridge University Press. Cambridge

[19] Yearwood J. L., 1997. Case-based Retrieval of Refugee Review Tribunal Text Cases. In A. Oskamp and R. V. De Mulder and C. van Noortwijk and C. A. F. M. Grtters and K. Ashley and T. Gordon (eds) *Legal Knowledge Based Systems. JURIX: The Tenth Conference*, pages 67-83.

[20] Zeleznikow, J. and Stranieri, A. 1995. The Split Up system: Integrating neural networks and rule based reasoning in the legal domain. *Proceedings of the Fifth International Conference on Artificial Intelligence & Law. ICAIL'95.* ACM Press. New York. pp185-194.

[21] Zeleznikow, J., and Hunter, D., 1994. *Building Intelligent Legal Information Systems.* Kluwer Law and Taxation. Deventer.