

Facilitating Meaningful Public Access to Primary Legal Information: Designing an Integrated Legal Environment

*Marcus Bornfreund & Phil Surette**

A. INTRODUCTION

In this chapter we, the authors, describe our experience building a tool for browsing statutes and other primary legal information in a rich, integrated electronic environment. This work builds upon an earlier project: a system for automatically extracting the structure inherent in the unstructured corpus of Canadian federal legislation, which is published in HTML format.

B. ENABLING FASTER, FREER ACCESS TO LEGAL INFORMATION

1) Introduction

In this chapter we briefly describe existing obstacles to the accessibility and usability of primary legal information currently available online. The main focus of the chapter is a description both of our progress thus far, and future plans to help address these problems, with a facilitative tool called the LawShare Browser.

2) What Is the Problem with How Legal Information Is Stored?

The structure of online legal information today reflects the limitations of the legal world's paper-based legacy. The foundational problem with the way legal information is currently stored is that the industry has simply carried forward the structures of a paper-based world to the digital environment. The following problems have been inherited:

* You can download the LawShare (ILE) Browser at <http://lawshare.ca/resources.htm>.

- 1) links between pieces of information are cumbersome and difficult to update;
- 2) information that could be organized in many different ways is stored in a single linear or hierarchical fashion; and
- 3) sharing of secondary information is time-consuming and slow.

Of course, paper is no longer the principal medium for storing legal information. We have large, private, searchable databases of caselaw, legislation, and secondary materials. However, these databases are little more than repositories of scanned print material combined with a search engine. This is an important first step but leaves much to be desired, as meaningful interaction is largely absent. Moreover, the existing structure in government-published legislation and caselaw is barely exploited; for example, links within a statute—and to revising acts—are rarely available, making it unnecessarily slow to navigate links between pieces of information.

a) Structured and unstructured legal repositories

An unstructured legal repository is a searchable database of largely non-formatted primary legal material. This definition is broad enough to capture normal legal web applications such as laws.justice.ca, CanLII, and commercial legal databases. At their most basic level, these applications marry a collection of primary legal materials with a search engine to allow searches for particular terms within statutes or caselaw. Statutes (or sometimes top-level sections) and cases are treated as basic or unstructured documents within the repository. Although they may also allow searches within statute or section titles, and may provide navigation to particular sections, very little of the structure of the underlying content is directly represented in the database. A defining characteristic of an unstructured legal repository is that it is stored in a presentation format (such as HTML or PDF) that presupposes that legal data will always be presented in the same way.

A structured legal repository is a legal repository of structured legal content. Structured legal content is represented using a data format such as XML, which exposes the relationships between parts, topics, sections, subsections, paragraphs, and metadata such as headings and marginal notes, but does not include presentation markup. The structured data supports the following key features:

- more precise linking (pinpoints);
- more precise searches; and
- different ways of organizing and representing the same data.

The capability to precisely link to pinpoints within a piece of legislation is a key enabler for effective annotation, topic construction, and concordance between sections and paragraphs of a statute across amendments. A structured legal repository will support a better, more accessible legal system.

b) Lack of sharing or reuse of legal information

Because there is no free, public, high-quality legal repository (structured or not) that is linked to a legal collaboration system, the research results and insights of lawyers and students are not shared or reused outside the confines of a law firm. Instead, the wheel is constantly reinvented—resulting in more time spent researching, with poorer results than if each successive researcher could build upon the previous researcher’s work.

Absent a shared repository of legal metadata, there will never be an authoritative, up-to-date source of information on a particular legal topic because legal information changes too rapidly for print-based secondary sources to be current. Every decision has the potential to change the law in unpredictable ways. Secondary legal materials provide the best resource here but always lag significantly behind the law and, due to the unfavourable ratio of the number of legal authors in a jurisdiction to the volume of law, generally address only the most common legal problems.

3) Why Should We Fix It?

The current system for interacting with primary legal information is inefficient. If you are lucky, you may already know of some secondary sources that can point you in the right direction. Or you can perform unstructured searches of existing caselaw, hoping to find references to the section. This is error-prone: you will need to search for terms that are used in the section, the exact text of the section (hoping the judges have quoted it), or the appropriate section number somewhere near the name of the act (keeping in mind that sections are regularly renumbered). The odds are good that you will miss a lot or be swamped by too many results.

The following subsections describe in detail reasons why we should not be satisfied with current, unstructured legal repositories. Some of the reasons are related to building in support for the kind of legal collaboration system that is needed to enable sharing and building legal knowledge, and some relate to more direct benefits.

a) Islands of information and inefficient linking

To understand the meaning of a section of a statute, it is necessary to know what other legislation it relates to, what pronouncements the judiciary has made on the section, and what secondary materials may apply. Legal information is highly interrelated. But legal information is not stored in a way that facilitates finding this related information; today, the user is generally given nothing better than *ad hoc* search strategies to get to this information. Reliable and pervasive linking is required to convert legal information from a series of islands into a highly interconnected cloud of information.

In some ways, the legal world already has an advanced linking mechanism in the form of citations. However, citations are complex and (except for the neutral citation format) publisher-specific. Much of the time, online versions of legislation and cases are not marked up to allow pinpoint referencing to work effectively. But the biggest problem in this area is that citations are generally not live; that is, they are not linked to an online version of the case or statute. Over time this has been improving, but richer linking structures are required. We need full-fledged meta-information, arranged by topic through a collaboration system, with bi-directional links between a topic and related primary and secondary legal material.

b) The granularity problem

The “granularity problem” is a term used within the artificial intelligence discipline to describe the question of knowing at what level of detail to represent knowledge. Legal publishers have not explicitly addressed this problem. Primary legal materials are packaged as statutes, sections, and judgments. The amount of information in each of these units is too large, or too *coarse*, to allow optimal access to the information.

For instance, while judgments provide the appropriate level of granularity for resolving a dispute, as legal instruments they are far too rough. Lengthy judicial decisions are routinely reduced by practitioners to the legal principles they represent. Once the decision is rendered it is these simple principles that inform the law. The rest of the decision is largely irrelevant outside the scope of the decision itself. Because decisions are stored in unstructured legal repositories without any metadata to filter them, the irrelevant information in caselaw creates a substantial and often overwhelming amount of noise in the search result.

c) Hierarchical representation of data

Legislation is designed in a top-down, hierarchical fashion. This reflects the structure of paper documents and means that statutes can only be

efficiently organized along one dimension. Current legal repositories exhibit the same limitations as the statutes themselves. These limitations can really only be overcome by superimposing an overlay of topic-centred meta-information on top of the legislation. This is what secondary legal materials have traditionally tried to do, though in a static way and in documents that are themselves often largely linear. What is needed today is something more powerful: a collaborative annotation system providing new ways of organizing and viewing legal data.

d) Unacceptable response time

Response times of over 300 milliseconds are enough to interrupt a user's train of thought. Because existing legal repositories lack sophisticated linking facilities, finding legal information suffers from time lags caused by inaccurate searching, analysis of search results, and refining of criteria. Minutes, hours, or even days can elapse when trying to determine the legislation or caselaw that informs a particular law. The process is inordinately slow compared to other information-rich domains. For instance, software developers have access to integrated development environments that allow them to navigate through programming information instantaneously, even though the amount of information they have to navigate is similar in size and scope to what legal practitioners navigate.¹

Because of these inefficiencies, legal research is slow. This results in poor access to law for legal professionals and laypeople alike, less well-informed legal decisions, and a legal process that is inefficient—with commensurate costs to society as a whole.

4) Who Cares?

There will be no improvement to legal information without a demand for it. Legal practitioners are both the primary beneficiaries and the serfs of the current system. If the system was more efficient, it might result in higher-quality legal decision-making and less drudgery for legal professionals. But it might also mean less legal work. “Cheaper legal services” is not a rallying cry that many lawyers will instinctively support. Similarly, the legal pub-

¹ For comparison, consider that the complete set of federal statutes and regulations, in XML format after conversion on laws.justice.gc.ca at the end of 2005, was 289 megabytes of data. The core API documentation for the popular programming language, Java, was about the same size—185 megabytes for JDK1.5.

lishers and database managers cannot be expected to independently facilitate significant liberalization of access to their “proprietary” information.

But there are some groups who should care, for example:

- Government: governments are presumed to take their duty to communicate their laws to citizens seriously.
- Citizens: citizens are governed by laws, many of which are complex and contextual, making it virtually impossible to understand, interpret, and apply them without both legal training and specialization. The public would benefit from more accessible primary legal information both directly, by being able to understand it themselves, and indirectly by improving the quality and efficiency of legal advice.
- Students: law students are struggling daily with the problems that stem from poorly structured legal information, and they should be more motivated than most to find a better way.
- Sole practitioners: a freely-accessible repository of legal information would help to level the playing field between sole practitioners and firms with access to (expensive) proprietary repositories and databases of in-house legal materials.
- Judiciary: more than anyone, the judiciary has an interest in ensuring that judicial decisions can be made quickly and accurately, with reference to current precedent.

5) The Solution: An Integrated Legal Environment

The solution to the limitations of the current legal repositories is to create a public, living repository of legal information that is easy to navigate and contribute to. This can be achieved through the development of an Integrated Legal Environment (ILE). The idea of an ILE is to try to build a tool that will provide legal practitioners with the level of tool support that is available for software development through Integrated Development Environments (IDEs) today. This means that you should be able to move between statutes, sections of statutes, relevant cases, and annotations almost instantaneously.

The following are key technical components of an ILE:

- a highly available, public, *structured legal repository*;
- a powerful *legislation browser* for navigating the repository; and
- a shared *legal annotation system* to support the creation of the legal metadata—topics, links, and notes that superimpose order on the mass of legal information.

C. DEVELOPMENT OF AN INTEGRATED LEGAL ENVIRONMENT

This section describes the current status and a longer-range plan for developing an ILE. The following issues are addressed:

- who will contribute to the ILE;
- the copyright regime under which the ILE will operate; and
- progress, plans, and a technical description of the ILE.

1) Who Will Contribute to an Integrated Legal Environment

An Integrated Legal Environment will be of limited value without a community of *contributors* or *annotators* willing to enhance the raw data in the structured legal repository. It is expected that people will contribute for the same types of reasons that motivate Wikipedia contributors, open source software developers, and even bloggers. Some possible reasons are

- to create a permanent record of one's research that is externally maintained and updated;
- to build an online reputation;
- to use a public forum to encourage (legal) reform;
- to share one's knowledge; and
- a philanthropic impulse to improve access to law.²

It is expected that the early adopters of the ILE will be law students, who can use an ILE as a way of making and sharing notes with one another.

2) Copyright Issues in an Integrated Legal Environment

An ILE will contain the following types of copyrighted information:

- federal legislation and caselaw; and
- contributed metadata in the form of annotations, categorization, link creation, and the like.

Federal legislation and case law can be freely reproduced if due diligence is taken under the statute appropriately named *Reproduction of Federal Law Order*.³ Because the ILE will convert federal law from one format (HTML)

2 See WorldLII, "Declaration on Free Access to Law," online: www.worldlii.org/worldlii/declaration.

3 Department of Justice Canada, "Reproduction of Federal Law Order (S.I./97-5)," online: <http://laws.justice.gc.ca/en/ShowDoc/cr/SI-97-5//en?page=1>.

to another (XML), care must be taken to ensure that the due diligence requirement is met. For instance, it may be that special permission will be required from the registrar of the applicable court to include headnotes.

Contributions made to the ILE in the form of annotations or other metadata will use a Creative Commons Attributions-Share Alike licence⁴ to ensure both free access to the content and to allow the content to be updated. A fully featured annotation engine will need to keep track of who contributed a particular piece of metadata so that both a chain of authority and (if the contributor desires it) attribution can be made for a particular contribution.

3) Development Plan for the ILE

The first component of the ILE that has been developed is the LawShare Browser, a legislation browser intended as a rich tool for exploring statutes. The intent is to provide smarter and faster statute navigation than is currently in use in web-based tools available both from governments and commercial sources.

The LawShare Browser is one piece of a long-range project aimed at providing an open source, free, shared legal development environment—similar to the collaborative, rich development environments that software developers currently enjoy with IDEs. This environment will couple rapid and intelligent navigation of legal materials with a rich knowledge-sharing mechanism. The following sections describe the various phases of the planned development process.

4) Phase 1: Adding Structure to Statutes (Completed)

The project must reach a certain level of usefulness before it will attract users and contributors. We have decided to target law students as a user-group for whom we can provide a useful product quickly. For this group, simply providing access to a comprehensive set of federal statutes that are properly linked and easily navigated will provide significant value to their studies.

The federal statutes at laws.justice.gc.ca are published in HTML format. They are not very searchable, have links only to the section levels, and have no internal links. The first goal of the ILE project was to create

4 See the licence terms at Creative Commons, “Attribution-Share Alike 2.5 Canada,” online: <http://creativecommons.org/licenses/by-sa/2.5/ca/>.

XML versions of these statutes. This phase was completed in 2006, with the complete collection of Canadian federal statutes and associated regulations being converted to an XML format that could be used by the LawShare Browser. The completed conversion consists of 9142 files containing 309 MB of data.

Since the time of conversion, the federal government has made slight modifications to the HTML format of its statutes, which may necessitate further changes to the conversion tool. Ultimately, governments should be convinced to publish structured versions of their laws, but this is a very long-range goal. In the meantime, subsequent converters will be simplified by using the following two-step process:

- 1) convert the published representation of the statute to text-only; and
- 2) use pattern-matching to identify the underlying structure.

This approach, which results in an intermediate text format, has the danger of potentially dropping some information when converting to text, but has the benefit that the same pattern-matching code can be used to parse legislations from all jurisdictions. The current converter uses a one-step process and tries to use cues from the HTML classes of the generated data to identify structure, but this process is too brittle in the face of changes to the presentation format in which the federal government chooses to publish.

The current conversion utility captures the following structures within a piece of legislation:

- containment relationships between parts, topics, sections, subsections, paragraphs, and clauses;
- the history of amendments to a particular section; and
- metadata such as keywords, titles, definitions, and marginal notes.

The work of creating links within statutes on this body of data remains to be completed and will never be fully automatable. For instance, when section 5 of the *Broadcasting Act*⁵ refers to “subsection 3(1),” it should be possible to follow that link easily. Many of these links can be created automatically based on imprecise, language-dependent, and regular expression-matching. Imprecision arises from the many different possible ways of describing a reference, language difference, and the context-sensitivity of the reference.

However, it is expected that the vast majority of intra-statute references will be able to be identified with good reliability. A fully integrated LawShare Browser will identify which links were created automatically, and

5 S.C. 1991, c. 11.

will allow online contributors to correct and expand on the automatically created links, as well as presenting a list of possible endpoints for links that are suggested by the browser.

The creation of the containment relationships mentioned above creates many benefits. Users can link or navigate to components of the statute at a very fine-grain (clauses) or coarse-grain (parts) level. This is important to allow annotations to be properly scoped. It also allows LawShare Browser to dynamically display or hide useful contextual information, such as the list of amendments that apply to a section, marginal notes, or ultimately comments on the selected piece of legislation that have been provided by members of the community.

Other benefits of making the containment relations explicit include:

- allowing the calculation of intelligent differences between versions of a statute;
- allowing a web service to deliver only the amount of data required by the client (for better responsiveness in an AJAX or similar application); and
- allowing the generation of tables of sections, forms, schedules, parts, or other logical groupings of the statute.

5) Phase 2: LawShare Browser (completed)

The objective of this phase was to develop a prototype legislation browser, called the LawShare Browser, which would be able to demonstrate some of the benefits of structured statutes. In this section we describe the nuts and bolts of how the LawShare Browser prototype was built and the rationale behind the choices that went into building it.

A number of technologies were considered for developing the statute browser. The following factors guided the choice of technology:

- 1) The statute browser needs to be cross-platform to reach the broadest possible audience. In particular, it should be supported on open source operating systems (Linux and other open source unices). Java-based (Swing or Eclipse-based) or web-based (HTML and XUL) applications meet this criterion.
- 2) The technology upon which the statute browser is based needs to provide a rich and responsive user experience. This rules out traditional (pre-AJAX) web-based applications that require a round trip to the server for every user interaction, which introduce overly long latencies between performing an action (such as expanding a tree node) and get-

ting a response. The statute browser must follow “good human-factors engineering principles,” which dictate that responses must occur within 300ms to avoid interrupting the user’s thought process.

- 3) The underlying technology needs to be easily deployed over the web. The model for the LawShare Browser is to download entire statutes and store them entirely locally, but in a full deployment the original statutes will be pulled from a central repository of “XMLized” statutes that will be made available over HTTP and continually updated.
- 4) The underlying technology must allow the statute browser to keep a large cache of information locally, including both cached statutes and local annotations of the statutes. This rules out unsigned Java applets or normal HTML applications.
- 5) It must be possible to develop the statute browser in a timely fashion.
- 6) The statute browser must be built on open, preferably open source, software.

XUL was chosen as the underlying technology of the LawShare Browser. It is the open source user-interface definition language that is used both to build the Firefox web browser and to deploy applications within Firefox. Because XUL applications run *within* a browser, they can take full advantage of the very rich document manipulation capabilities of a full-fledged web browser. XUL applications can be installed *within* a browser and can take full advantage of the very rich document manipulation capabilities and, as trusted applications, they can also get access to the file system.

An annotated screenshot of the LawShare Browser is shown in Figure 1. Features of the current version of the LawShare Browser include:

- A side-by-side *hierarchical* view of a piece of legislation and an HTML/CSS *content* view of the associated text—both views are always synchronized.
- An optional display of amending legislation for particular sections.
- A direct pinpoint search: enter a pinpoint and the browser will take you directly to the relevant part/topic/paragraph/section.
- A structured search within text, titles, marginal notes, or definitions.

Some features that are yet to be added include:

- integration with an annotation engine;
- integration with a back-end repository of legislation (the current implementation is file-based); and
- the ability to generate a correctly formatted citation for the selected section or paragraph (a relatively simple feature to add).

Figure 1: The LawShare Browser

The hierarchy view provides an outline view for exploring the legislation and also provides context for what is displayed in the content view.

The content view uses faint dotted lines and indentation to clearly demarcate the scope of a part, topic, section, or paragraph.

Citation searches provide direct access to known sections by simply typing in the pinpoint, e.g. S14.2(2)(b).

Text searches are structured searches through the statute. In this case, the Copyright Act was searched for all occurrences of "moral rights" that appear in titles, definitions, or paragraph text. Occurrences of "moral rights" that appear in marginal notes have not been returned.

Sections that have been amended over time have clickable headings. In this example, the user has clicked the s. 14.2 heading to reveal that the section was last amended in 1997. There is a direct link to the amending act.

6) Phase 3: Deployment of Structured Legislation into a Shared Legal Repository

This phase has yet to be completed. The goal of this phase is to design a highly available, remotely accessible legal repository.

A legal repository needs to support the following features:

- Full indexing of legal text, headings, and metadata such as marginal notes;
- Multi-language support;
- Efficient search capability;
- The ability to store multiple versions of the same statute to support point-in-time searches;
- The ability to maintain and navigate the structure of legislation and caselaw;
- Storage of the following types of legal material:
 - » Legislations
 - » Caselaw
 - » Metadata about jurisdictions, courts, judges, *et cetera*; and
- Integration with an annotation system.

Content repositories are an emerging technology that is well-suited to support the above features. For example, JSR-170⁶ content repositories have built-in text indexing capabilities and native XML support, including a query engine that is based upon XPath. The content repository sits on top of a file system or a database and adds in the kind of support for text indexing, searching, and navigating hierarchical data that would need to be built in any event if a flat file system or database was used directly. XML databases are a similar maturing technology that are also a good match and could possibly be used in conjunction with a content repository.

A high-level design of the repository has been proposed based on the capabilities of a content repository, which is a kind of hierarchical database. The core legal information in the repository will be organized at the following levels:

- 1) Jurisdiction;
- 2) Lawmaking body (a court or a legislature);
- 3) Legal document (decision or legislation);
- 4) Document version (in the case of legislation, each amendment results in a new version of the legislation);
- 5) Language; and
- 6) The internal organization of the legal document itself (e.g., opinions, sections, and/or paragraphs).

6 See further at Java Community Process, “Community Development of Java Technology Specifications,” online: <http://jcp.org/en/jsr/detail?id=170>.

The following sections examine some aspects of the proposed structured legal repository in detail.

a) How to handle different languages in the content repository

The problem of how to represent multiple languages must be carefully addressed when designing the content repository. Multiple languages mean that you will have multiple versions of various units of legislative information (legislation and decisions). This does not sit easily with a hierarchical layout. There are two approaches that are generally used to address the issue of multiple languages in a hierarchical repository:

- Early splitting: Parallel hierarchies are created, beginning at the root of the repository. This is the approach that the Government of Canada uses for its legal repository.⁷
- Late splitting: A single hierarchy is used, with branching between languages occurring only at the leaf nodes.

Figure 3 (page 144) exemplifies the differences between these two approaches. The clouds indicate the point in the hierarchy at which duplication begins.

For the structured legal-content repository we have decided to follow a late-splitting strategy. It has the following advantages:

- The hierarchical relationship between nodes is preserved as far as possible.⁸
- There is less duplication of information.
- There is less need to keep separate branches in sync.
- It is a more flexible model for allowing the addition of new languages.⁹

7 See French: Ministère de la Justice Canada, “Site Web des Lois du Canada,” online: <http://laws.justice.gc.ca/fr>, and English: Department of Justice Canada, “Justice Laws Web Site,” online: <http://laws.justice.gc.ca/en>. For these websites there is a common homepage which directs you to either the English or the French website, both of which are for all intents and purposes entirely independent.

8 This avoids difficulties that can arise where only one translation is available for a particular node. In the early splitting strategy, for example, if the *Patent Act* was not available in English, and the *Copyright Act* was not available in French, then the only way to discover that full set of facts in the federal jurisdiction would be to perform a merge of the various translations of the Acts.

9 For instance, with early splitting, if you wished to add Inuit as a third language for federal Canada, you would need to create a new hierarchy containing all Inuit translations at once. With late splitting, you can just add Inuit translations as they become available.

The Government of Canada can achieve its goals with an early-splitting strategy because federal legislation is required by law to be published simultaneously in English and in French. However, the structured legal-content repository is intended to be applicable to all jurisdictions, and some jurisdictions may provide different levels of support for different languages, which means that translations into some languages will not always be available. Late splitting is more tolerant of translation gaps.

The final question that remains is: Given that we intend to use late splitting of languages, precisely how late should the splitting occur? The obvious place to split is at the level of individual pieces of legislation or decisions. This is the policy that we intend to pursue. However, another strategy would be to push the splitting even later. For instance, a given statute will have precisely the same number of parts, sections, and subsections in both languages (to allow interoperability between languages), so the split could be deferred all the way down to leaf nodes within the statutes. Because this is even later splitting, this is an attractive strategy for all the reasons listed above.

Nevertheless, we have chosen not to utilize late splitting because it is extremely intolerant of mismatches between different language versions of the same statute, which are likely to occur due to typographical errors in the source statute, or from errors in the parsing process used to convert source statutes into structured (XML) statutes.

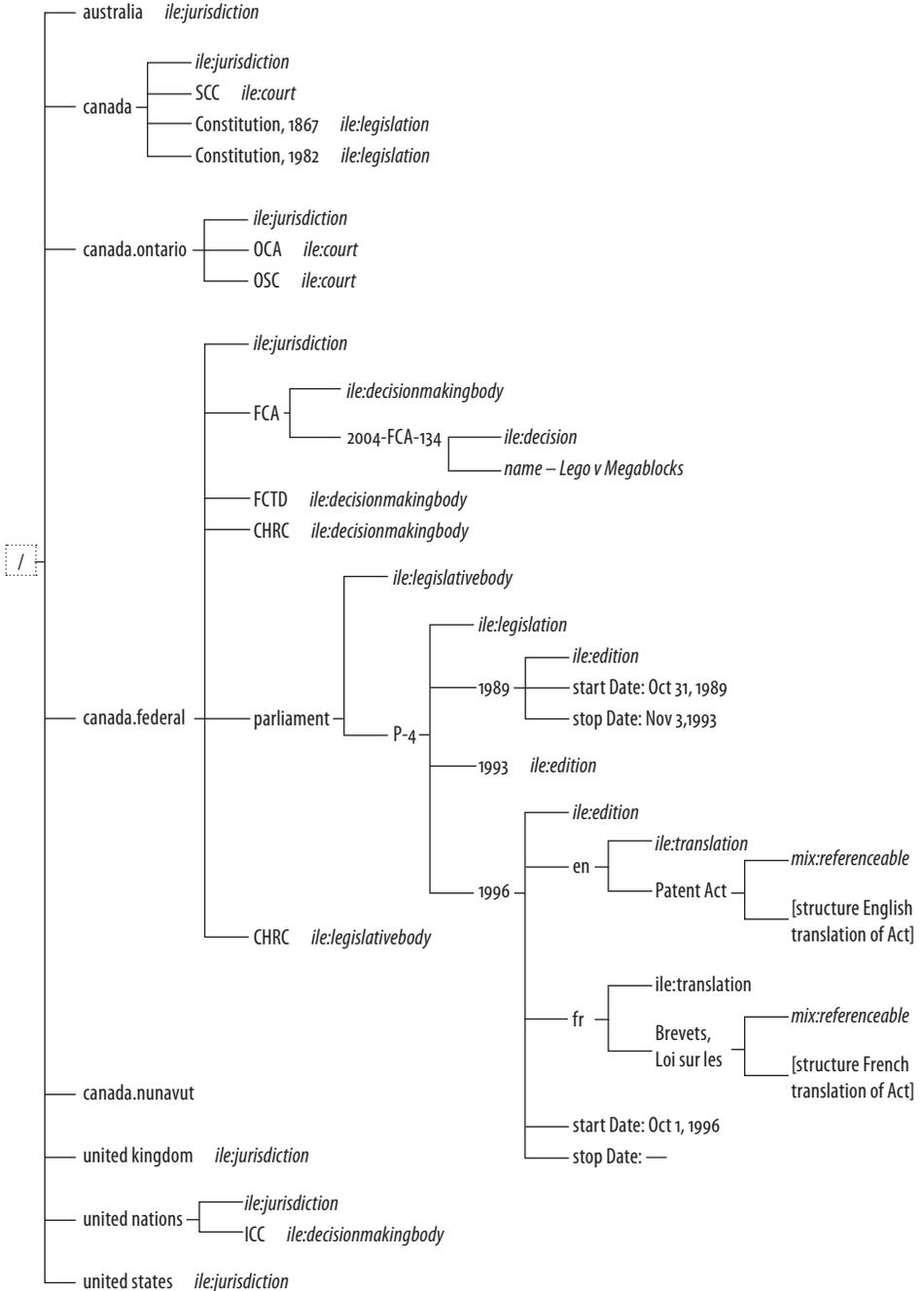
b) How to design jurisdictions within the repository

One of the most obvious constraints on searches within the structured legal-content repository will be to limit a search to relevant jurisdictions. This division is so familiar that some commercial repositories store material from different jurisdictions in separate databases.

Although it is obvious at first glance that jurisdictions are hierarchically organized within nations, there are some subtle aspects to jurisdictions that need to be handled carefully:

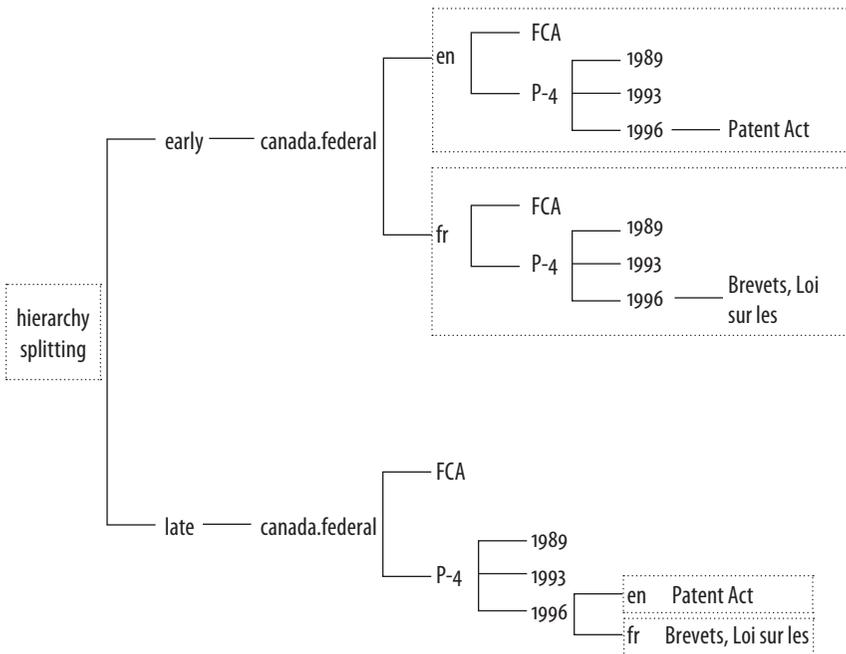
- Jurisdictions come and go over time—for example, the Canadian territorial jurisdiction of Nunavut is relatively new and the jurisdictions of upper and lower Canada have evolved into the provinces;
- Jurisdictions can merge or split—how do we represent the relationship between Nunavut and the Northwest Territories (from which Nunavut emerged)?; and
- Jurisdictions can move between national units—for example, Newfoundland joined the Canadian federation in 1949, and Czechoslovakia split into two jurisdictions in 1993.

Figure 2: Hierarchical Position of the Patent Act within the Content Repository



We think the correct approach to these questions is to not answer them at all. While jurisdictions appear to be stable, hierarchical structures at any point in time, over time they merge, split, and disappear. In order for the content repository to remain accurate, it should store jurisdictions as a flat structure. In other words, the repository should not explicitly represent the relationship between Nunavut and the Northwest Territories; this will be left to the realm of annotations.

Figure 3: Early vs. Late Hierarchy Splitting to Support Multiple Hierarchies



Jurisdictions will be represented within the structured legal-content repository as a completely flat list at the root of the repository, using a dotted naming convention (similar to that used for domain names) to group jurisdictions. Figure 4 shows the difference between the hierarchical approach and the flat approach to jurisdictions that we intend to pursue.

Note that although there is an explicit representation of a nation in the flat model, it will contain very little. For instance, in Canada the only objects that truly belong to the jurisdiction called “Canada” would be constitutional documents and the Supreme Court of Canada; these are in their

own jurisdiction that supersedes all other jurisdictions in Canada, but this is not represented in the repository other than through the hint that the dotted naming convention provides. The Parliament of Canada, its legislation, and federal courts all reside within the Canadian federal jurisdiction.

c) Representing courts and decisions

In a complete, structured legal-content repository, decisions should be linked to the decision-making bodies that make them. A special node type, “ile:decisionmakingbody,” will represent courts, tribunals, and the like. Decisions will be represented by the “ile:decision” node type and will be children of ile:decisionmakingbody nodes.

Another important component of decisions is the group of judges, tribunal members, or arbitrators who made the decision. These will be represented within the repository as children of an ile:decisionmakingbody with the type “ile:decisionmaker,” and these nodes will contain meta-information about the decision maker, indicating the periods in which they were active in the decision-making body, as well as links to their participation in other decision-making bodies.

Individual decisions will contain references to the list of ile:decision-makers that participated in the decision. Decisions will also have links to earlier and later decisions in the case history, where available.

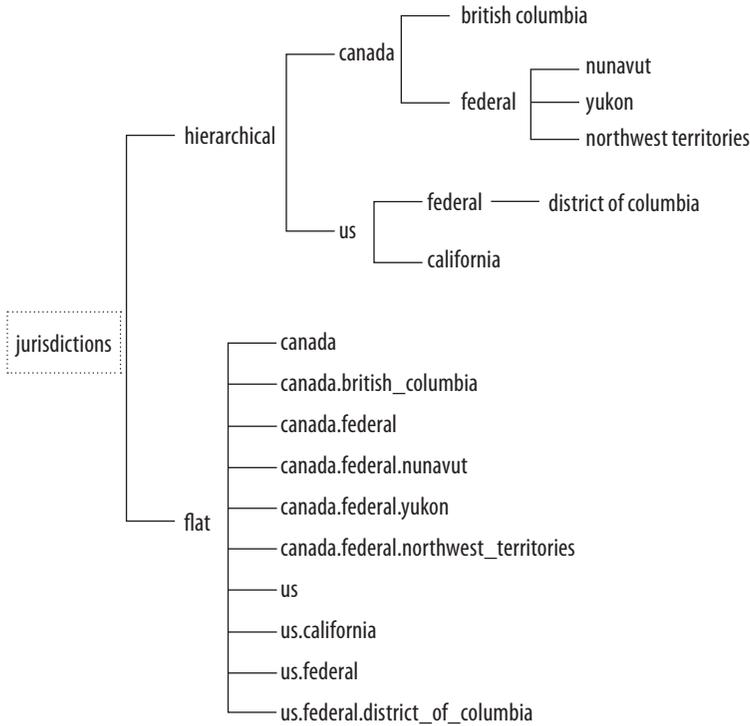
d) Representing legislation and legislation-making bodies

The relation between legislation and legislation-making bodies is similar to that between decisions and decision-making bodies. This relationship is visible in Figure 2 (page 143).

e) Interfaces to the shared legal content repository

The repository will be accessible as a web application and as a web service. The web application will be used by end-users from a standard web browser. The web service will be available to rich clients such as a statute browser, and will be designed to be easily integrated with AJAX applications. The web service will also be available for other custom applications such as data-mining applications.

Figure 4: Hierarchical Representation of Jurisdictions vs. the Proposed Flat Model



7) Phase 4: Annotations

Annotations and topics or articles (similar to Wikipedia articles) will provide the key means of providing both links between laws and non-obvious perspectives on how different laws interrelate.

It is currently envisioned that annotations will not be stored directly in the legal repository. Every node in the legal repository will be uniquely identifiable with a structured URL that will contain the same type of information currently available in the neutral citation format, with implementation-dependent extensions. The annotation engine will be a separate database that links to the legal repository using these identifiers. The design of the annotation repository must make it possible to quickly identify which annotations apply to a piece of law.

Annotations should be supported anywhere in the repository. The annotation system has to support the following:

- links to the annotator;

- date of annotation;
- links to external resources;
- an annotation history; and
- codified rules for who can contribute annotations (the initial model will be informed by the social rules of Wikipedia).

8) Phase 5: Structured Caselaw

This phase will structure caselaw in much the same way as Phase 1 added structure to statutory materials. The targeted caselaw is that of the Supreme Court of Canada, since it is both the most widely applicable jurisprudence in Canada, and the most complicated jurisprudence. Because of the complexity of having multiple opinions and references interwoven in a single judgment, the Supreme Court judgments will benefit the most from being structured.

9) Future Developments

Other functionality that may be added includes:

- The ability to navigate amendments to a statute through time, preferably through a diff-based mechanism;
- Providing quick navigation between equivalent English and French versions of legislation; and
- Advocating the integrated legal environment approach for adoption by third parties (such as the Government of Canada itself).

D. SUMMARY

In this chapter we have briefly described our plans to build an integrated legal environment. A working version of the first application, the LawShare Browser, is already a useful tool and, with a little more work, will be a compelling application in its own right. However, it is not until it is integrated with a web service that provides easy access to structured versions of all federal statutes, together with a common annotation engine, that the full power of an integrated legal environment will be measurable.

The prototype LawShare Browser demonstrates some of the benefits of XML-structured statutes by providing access to a select set of federal statutes that are properly linked and easily navigated. It is available for download from www.lawshare.ca/resources.htm.