PLINTH:
Integrating Hypertext, Semantic Nets and Rule-Based Systems in an Expertext Shell for Authors and Readers of Regulatory Documents

Andrew Casson

AIAI-TR-142

September 1993

Published in: Mayfield, J. and C. Nicholas (eds.),
Proceedings of Workshop on Intelligent Hypertext
CIKM '93, Arlington VA., November 1993

Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

PLINTH: Integrating Hypertext, Semantic Nets and Rule-Based Systems in an Expertext Shell for Authors and Readers of Regulatory Documents

Andrew Casson
Artificial Intelligence Applications Institute
University of Edinburgh

September 1993

Introduction

This paper describes current research, in collaboration with the Building Directorate of the Scottish Office, concerned with the design and implementation of an authoring and information retrieval tool for technical documents, with particular application to the *Building Standards Regulations for Scotland* and the associated *Technical Standards*. The system we are building, called PLINTH (the **Pl**atform for **Int**elligent **H**ypertext) is an expertext[1, 2] shell which combines a variety of established representation schemes into a new, integrated architecture, which not only supports the creation and manipulation of standards documents in hypertext form, but also manages the design rationale underlying documents, and provides facilities for intelligent guided browsing and consultation of the texts. The paper explains the way in which the diverse representations have been combined, and gives an example of the system in use.

The problem

Both hypertext and rule-based systems are well established individually as tools for work on regulations and standards[3]. Hypertext systems provide structured access to technical documents, and can maintain and cross-reference large bodies of text, diagrams and tables. Rule-based expert systems exploit the inherently rule-oriented nature of regulations. However, both approaches have limitations:

Cognitive overhead: Hypertext has the well known problems of disorientation and digression, where readers fail to maintain a mental picture of their location and path in large networks, and get side-tracked or lost. Graphical displays of networks help greatly here, but for very large, highly interconnected documents even these can be difficult to untangle mentally.

Expressiveness: Expert system rules are limited in expressiveness compared with texts and diagrams. This is especially true with newer, less prescriptive regulatory documents where requirements may be expressed using terms like 'adequate means' and 'sufficient provision', as well as in precise weights and measures.

Parsimony: Conventional expert systems must reduce *all* knowledge to rules, whether or not this is appropriate. For example, although following the logical dependencies *between* clauses in regulatory documents is difficult, the *individual* clauses are usually expressed in clear, concise language that is quite easy to understand. This distinction can be exploited by focusing the system's intelligence on ensuring the correct and efficient use of documents, and trusting the user's intelligence to interpret their contents.

Maintenance: Creating large rule-bases for regulations, and maintaining the rules as the regulations are revised, is a time-consuming process requiring skilled programmers (this is partly a consequence of the **parsimony** problem).

Design rationale: Neither conventional hypertext nor rule-based systems are well suited to handling the *design rationale*[4] underlying technical documents. A substantial document like the Scottish Building Standards represents only the final product of a lengthy authoring process involving research, design, consultation and revision. The published text contains just a fraction of all the knowledge and data collected and used in this process, which can be characterised as design rationale (DR). A system intended to assist the whole authoring process must therefore support the interactive creation and manipulation of DR structures during the authoring process. It should also provide easy access to archived DR for authors' reference when documents are subsequently revised.

The PLINTH solution

Firstly, PLINTH augments hypertext networks with semantic properties. Types can be assigned to nodes and links to mark the function of the nodes in the network and the structural, logical and rhetorical relations between them. The user defines different network types by specifying the node and link types that can be used and the syntactic properties which constrain their use (*e.g.* the *mapping* of each link type which defines the types of nodes it may connect). This has several uses:

- The user can create, manipulate and browse **design rationale** using hypertext network types which conform to node-and-link based DR models such IBIS[5, 6], PHI[7] and DRL[8].
- Text nodes in document networks can be associated with nodes in their underlying DR structures by connecting them with ordinary hypertext links.
- The system can graphically display multiple structural, logical and rhetorical *views* of a single document network by activating or suppressing different sets of node and link types, thus reducing the perceived complexity of the network and the reader's **cognitive overhead** of understanding it.

In addition, nodes can be supplemented with slots which contain high-level representations of their meaning and content (*cf.* headed record expertext[9]).

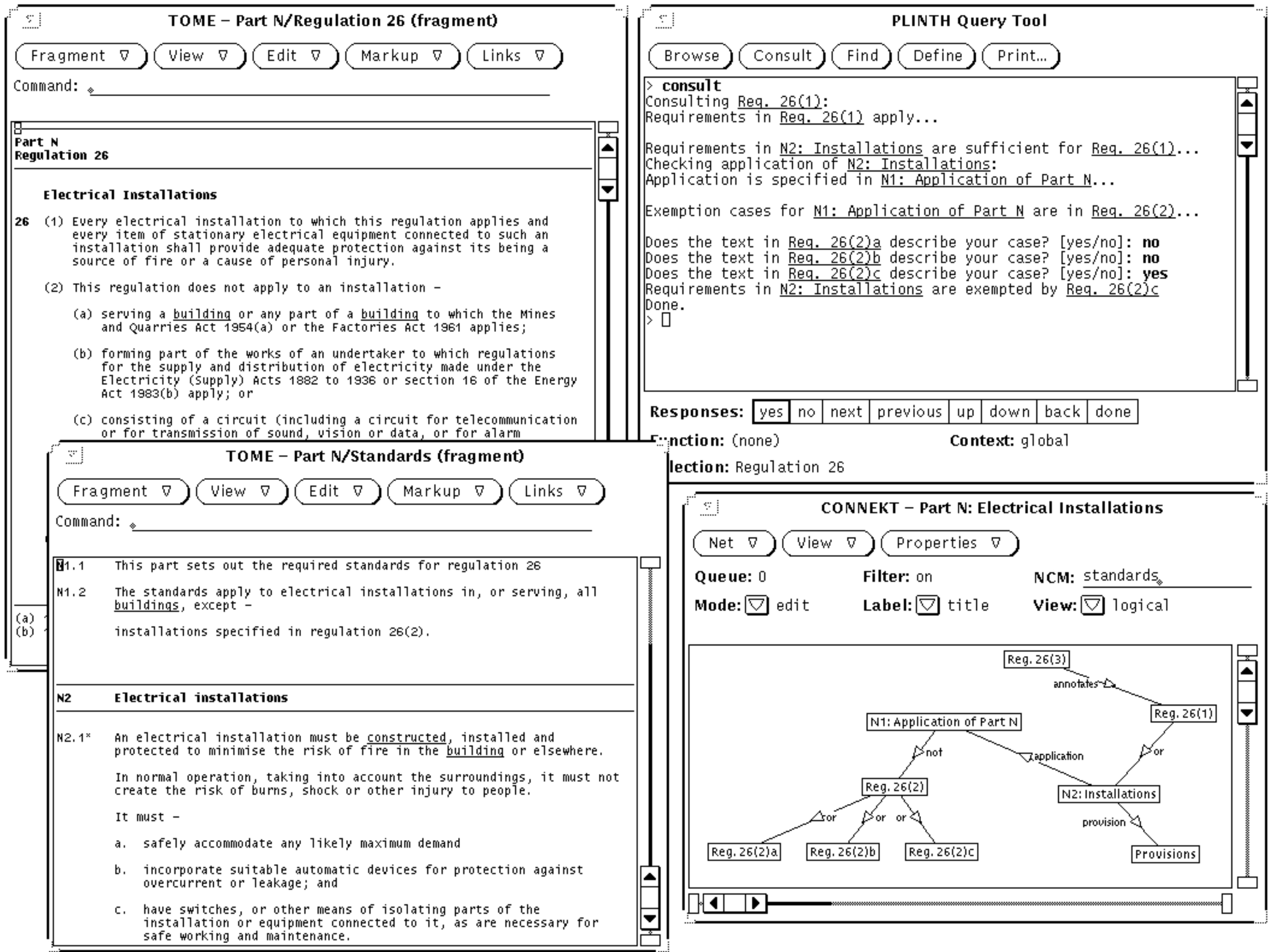
PLINTH then supports this augmented hypertext with rule-based intelligent navigation, to interactively compute a customised path through the document for the reader, based on the semantic properties and directed by commands and queries. This reduces the **cognitive overhead** problem even further. The author can write sets of navigation rules to perform different text retrieval or consultation functions corresponding to different views of the network. Here are some examples:

Free browsing: Here the navigator follows basic structural links like `next` and `part-of` in response to commands from the reader like *next*, *previous*, *up*, *back*, and so on, displaying the text of each node visited. Where a DR node is attached to a document node, the navigator can dip into the DR network, and a different set of browsing rules for exploring this type of network will be activated.

Consultation: The consultation facilities of conventional expert systems for regulations are achieved using navigation based on logical typing of nodes and links, *e.g.* `requirement`, `scope`, `necessary`, `sufficient`, `applies-to` and so on. In this case the navigator follows the links in logical order, assigning truth values to nodes by asking the reader whether or not the conditions expressed in the node text are met. These truth values are combined according to the link types in order to decide which requirements apply and advise the reader on how to satisfy them. This method alleviates the **expressiveness**, **parsimony** and **maintenance** problems:

- The texts of individual clauses need not be expressed as rules. There is no **expressiveness** problem with high-level functional concepts like 'adequate' and 'sufficient' which are difficult to represent, neither is there a **parsimony** problem of having to formalise clear, concise statements which do not benefit from it. However, it *is* sometimes useful to be able to break down clauses further, *e.g.* to do a calculation expressed in the text as a table or formula. For this purpose, PLINTH allows local rules to be attached to nodes and activated when the node is visited;
- Different documents with the same node and link types can use the same set of basic consultation rules (augmenting them with local rules where necessary), thus reducing the size and complexity of the rule-base and the problem of **maintenance**. Once those rules are written, an author without programming skills can write a new document and 'program' the required logical breakdown simply by assigning types to the nodes and inserting links between them.

Design rationale processing: The process of building a DR network can be supported by navigation rules which not only traverse existing parts of the network but also allow the author to extend it in a principled manner.



An example

The diagram above shows a snapshot of the screen during an actual session with the PLINTH system, where the reader is consulting Part N of the Scottish Building Standards. There are two text windows on the left, and the query tool and a graphical network view are on the right. Each part of the Standards has been marked up as a separate hypertext document using TOME, the Text Object Markup Editor, which is built on top of CONNEKT, the CONstrained NEtworK Tool.

The Part N network consists of 14 nodes and 30 links in total, but for logical consultation only the nodes and links currently displayed in the network view are actually relevant.

To begin the consultation the reader clicks on a node in the network view, in this case the node `Reg.26(1)` which is the logical starting point for Part N (the text of this node is the first clause in the upper TOME window). The reader then clicks the **Consult** button on the query tool to set the rules in motion.

The navigator follows the logical links `and`, `or`, `not` and `application` down through the net of `requirement` and `scope` nodes, explaining its reasoning as it does so, until it reaches the nodes corresponding to the three clauses of `Reg.26(2)` which contain exemption conditions for the standards in Part N. It then asks the reader whether any of these clauses is satisfied, highlighting each one in its TOME window in turn. In this example the third one is satisfied. The navigator then works back up through the net, inferring from the logical structure that the standards do not apply as an exemption condition has been found. The reader is informed of this and the process stops.

If the user answers **no** to all three questions, then the navigator infers that the standards in node `N2: Installations` do apply, and this is displayed and highlighted. There is no need for any further breakdown of the standards into rules as the text is quite clear in itself (see the lower TOME window in the diagram).

Summary

PLINTH supports authors and readers of technical documents. It allows authors to:

- construct semantically augmented hypertext networks for the documents and their design rationale;
- write rule-based intelligent navigation and consultation functions

in order to assist the reader in using the documents effectively and correctly. PLINTH allows readers to:

- display different structural, logical and rhetorical views of a document and manually select nodes for reading;
- explore the design rationale behind a document to better understand its purpose; and
- browse and consult documents aided by the intelligent navigation functions supplied by the authors.

Although it was designed initially for work on regulatory documents, PLINTH has great potential in many areas of technical documentation, for example:

- other types of regulatory and legal texts, for example British and ISO Standards, contracts etc.;
- instruction manuals or trouble-shooting guides, which lead the reader through a step-by-step process where each action depends on the outcome of previous actions;
- computer-assisted learning (CAL) material, where the information to be conveyed is basically textual, but the system has to present it in a way that responds intelligently to the reader's understanding and progress.

Acknowledgements

The PLINTH project at AIAI is currently funded jointly by the UK Science and Engineering Research Council (SERC Grant GR/J59197) and the Scottish Office. This follows on from a collaborative three year SERC project (Grant GR/H51477) by the University of Edinburgh (Dept. of Artificial Intelligence, and EdCAAD), Leeds University (Dept. of Civil Engineering), Queens University Belfast (Dept. of Architecture and Planning) and the Scottish Office.

References

[1] Diaper, D. and R. Rada (1989). "Expertext: Hyperising Expert Systems and Expertising Hypertext", in *Proceedings of the Conference on Hypermedia/Hypertext and Object Oriented Databases*, December 1989, pp. 123–152. Unicom Seminars Ltd.

[2] Rada, R. and J. Barlow (1989). "Expert Systems and Hypertext", in *Knowledge Engineering Review* 3, pp. 285–301.

[3] Kähkönen, K. and B.-C. Björk, Eds. (1991). *Computers and building regulations*, VTT Symposium 125, Espoo, Finland, May 1991.

[4] Lee, J. and K.-Y. Lai (1991). *A Comparative Analysis of Design Rationale Representations*. Center for Coordination Science TR# 121, MIT, May 1991.

[5] Kunz, W. and H. Rittel (1970). *Issues as Elements of Information Systems*. Report S-78-2, Institut für Grundlagen der Planung, Universität Stuttgart.

[6] Conklin, J. and M.L. Begeman (1988). "gIBIS: A Hypertext Tool for Exploratory Policy Discussion", in *ACM Transactions on Office Information Systems*, 6(4), October 1988, pp. 303–331.

[7] McCall, R.J. (1988). "MIKROPLIS: A Hypertext System for Design", in *Design Studies*, 10(4), October 1988.

[8] Lee, J. (1991). "Extending the Potts and Bruns Model for Recording Design Rationale", in *Proceedings of the International Conference on Software Engineering*, Austin TX, 1991.

[9] Diaper, D. and M. Beer (1990). "Headed Record Expertext and Document Applications" in *Proceedings of Hypertext Update*, Unicom Seminars Ltd, London.