Artificial Intelligence Applications Institute
AIAI — Technical Data Sheet

PLINTH: An Expertext Shell for Authors and Readers of Technical Documents

What is PLINTH?

PLINTH (the **Pl**atform for **Int**elligent **H**ypertext) is an *expertext shell*. It integrates the technologies of hypertext, rule-based expert systems and semantic nets to provide tools and intelligent support for the authors and readers of regulations and other technical documents.

Why is it needed?

PLINTH was originally designed to support the work of the Building Directorate of the Scottish Office on the Building Standards Regulations for Scotland and the associated Technical Standards.

Both hypertext and rule-based systems are well established individually as tools for work on regulations and standards. 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.

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 providing support for handling the *design rationale* underlying the technical documents, *i.e.* the data from research, discussion and consultation that goes into their production. The ability to structure and manipulate this information easily is very important for authors.

How does PLINTH solve these problems?

Firstly, PLINTH augments hypertext networks with semantic properties. The author can assign types and slots to nodes and links to mark the function of the nodes in the network and the structural, logical and rhetorical relations between them. This allows the author to:

- create, manipulate and browse **design rationale** (DR) using hypertext networks based on typed node-and-link based DR models like IBIS and DRL;
- connect document nodes to their underlying DR structures with ordinary hypertext links;
- show multiple structural, logical and rhetorical *views* of a single document 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.

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 for each view 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 browsing rules for exploring this 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.

What does PLINTH provide?

PLINTH provides tools and functionality to support *all* stages of the production and use of technical documents. Using PLINTH the author can:

- create and manipulate design rationale nets to organise the research data and design discussions on which the document will be based, experimenting with different DR models as required;
- use DR-specific net traversal rules to assist the process of building up DR nets;
- create hypertext nets for the documents themselves;
- edit the text of nodes, include bitmapped diagrams, and create hypertext anchors between nodes, using TOME, the *Text Object Markup Editor*;
- associate types and slots with nodes and links;
- classify and index text fragments using domain specific schemes (*e.g.* CI/SfB for buildings);
- write sets of navigation rules for the various kinds of browsing, consultation and text retrieval of regulations;
- output documents in PostScript format;

and the reader can:

- explore documents manually by clicking on nodes in a constrained graphical network view;
- browse and consult documents assisted by the intelligent navigation rules, directing and informing the navigator using the PLINTH Query Tool to enter commands, queries, and responses.

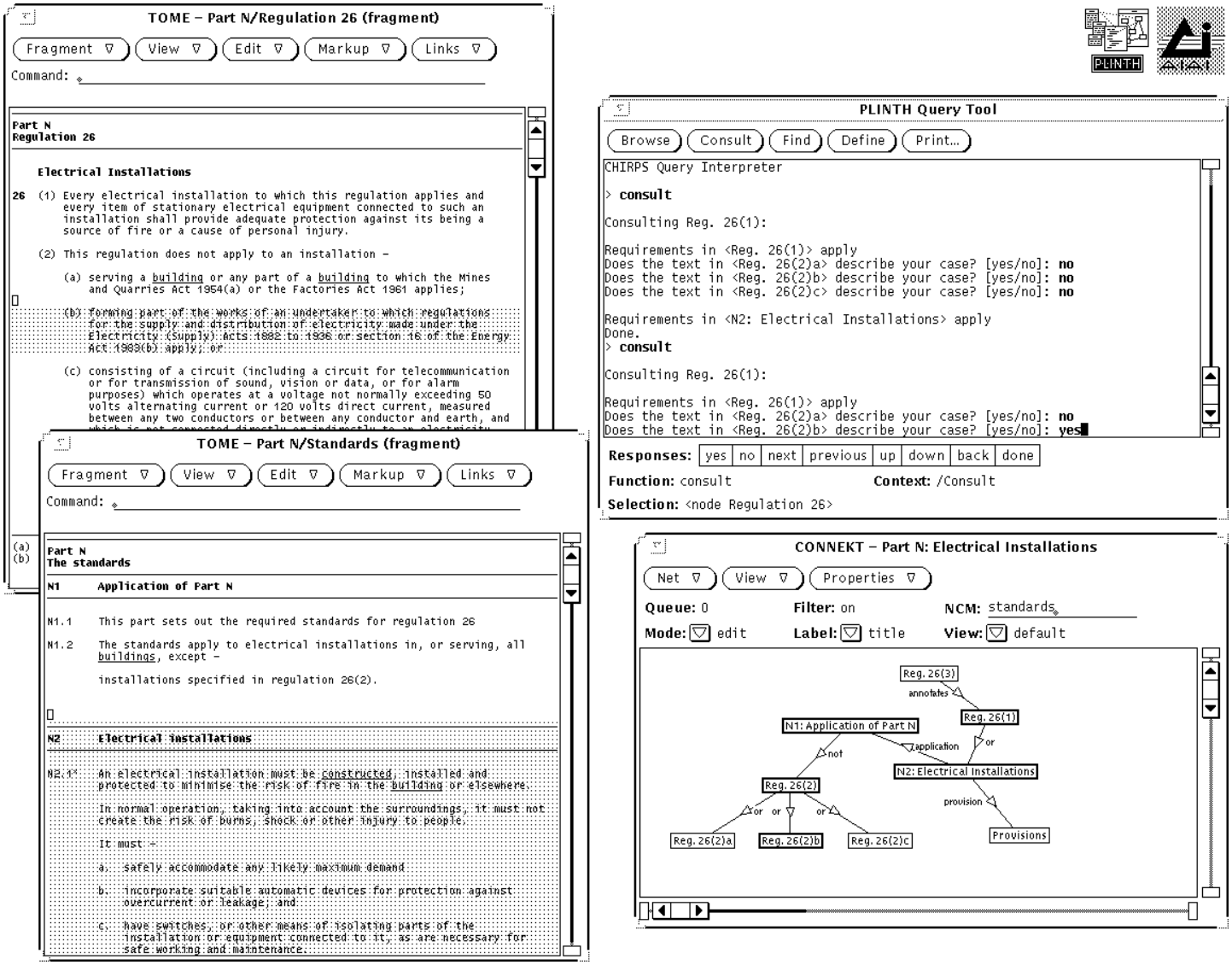

Figure 1: Consulting regulations using the PLINTH system

An example scenario

The diagram above shows a snapshot of the screen during an actual PLINTH session, where the reader is consulting Part N of the Scottish Building Standards. There are two TOME text windows on the left, and the query tool and a network view are on the right.

Each part of the Standards has been marked up as a separate hypertext network using TOME. The Part N network consists of 14 nodes and 30 arcs in total, but for logical consultation only those nodes and links currently displayed in the network view are relevant.

To begin the consultation the reader clicks on a node in the network view, in this case `Reg.26(1)` which is the logical starting point for Part N (you can see the text of this node as 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 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 none is, so the answer is **no** in each case. The navigator then works back up through the net, inferring from the logical structure that the standards

apply as no exemption condition has been found. The reader is informed of this and the requirements node `N2:Electrical Installations` is displayed (see the lower TOME window in the diagram) and highlighted. There is no need for any further breakdown into rules as the text is quite clear in itself.

Now suppose that, instead of answering **no** to each of the three clauses, the reader answers **yes** to the second (`Reg.26(2)b`), as is shown in the second consultation in the query tool window. At this point the navigator knows that one exemption condition is satisfied, and it will not go on to ask about the rest, but simply stop with the message:

```
Requirements in <N2:Electrical Installations> exempted by <Reg.26(2)b>
```

Summary

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

- construct semantically augmented hypertext networks for the documents and their associated 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.

Implementation status

The current version of PLINTH is a research prototype implemented in POPLOG Pop11 on Sun SPARCstations, using the X Toolkit and the Sun OLIT 2.0 Open Look widget set. In the next year of the PLINTH project we plan to investigate the production of PC version of the system.

Much of the future development of PLINTH will be guided by the results of its deployment in the Scottish Office. However, PLINTH has great potential in many areas of technical documentation:

- other types of regulatory and legal texts, for example British and ISO Standards, Acts of Parliament, 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 Science and Engineering Research Council (SERC Grant GR/J59197) and the Scottish Office. This follows on from the collaborative three year SERC project 'Intelligent Authoring and Information Systems for Regulatory Codes and Standards' (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.

Contact

For further details of PLINTH please contact us:

AIAI
The University of Edinburgh
80 South Bridge
EDINBURGH, EH1 1HN
United Kingdom
Tel: +44 (0)31-650 2732, Fax: +44 (0)31-650 6513
Electronic Mail: `AIAI@ed.ac.uk`