

RCAHMS Demonstrator Report

Kate Byrne, Steve Conway, Claire Grover, Amy Isard & Colin Matheson

19th May, 2005

1 Introduction

This report describes the work done by Edinburgh University's Institute for Communicating and Collaborative Systems (ICCS) for the Royal Commission on the Ancient and Historical Monuments of Scotland (RCAHMS) to produce a language generation demonstrator based on RCAHMS' data. The report partly fulfils the requirements set out in the original proposal (Matheson & Isard, 2004) as it contains reports on scalability and on the potential use of information extraction in this context.

2 Background

The background to this project is mainly the work done by Edinburgh University on the Ilex and M-PIRO projects. The Ilex system produces tailored descriptions of museum objects, altering the descriptions depending on factors such as what the user has seen before, and on the general type of user (descriptions for adults, for example, can be different in both form and content from those for children). The M-PIRO project extended the approach and added a multilingual element, producing descriptions in English, Italian, and Greek from the same underlying database, and also developed interactive tools for authoring the database.

The potential for using such a system with the RCAHMS data is clear; visitors can be provided with tailored descriptions which are aimed specifically at a particular user type, which do not repeat earlier material, which can compare and contrast items with previous information, and which are (at least theoretically) in any language. The demonstrator project set out specifically to investigate the use of the existing software in producing a software demo, and to report on the potential use of information extraction techniques to create language generation databases semi-automatically from existing databases. The project also looked into the question of how viable it would be to scale the existing software up to cover the many thousands of entries in databases such as RCAHMS'. Another important aspect was the agreement that ICCS would create a new web front end for the system.

This work was divided into 4 workpackages:

- WP1: Authoring of RCAHMS data in M-PIRO format
- WP2: Writing a front end application
- WP3: Assessing scalability
- WP4: Investigating information extraction

These are described in detail below.

3 Workpackages

3.1 WP1: Authoring RCAHMS data

In authoring a new database based on RCAHMS data, we identified four main sub-tasks:

1. Identify a suitable subset of the RCAHMS records, taking into account which aspects of the records are closest in format to the current M-PIRO data.
2. Create a type hierarchy covering the selected objects, using the M-PIRO authoring tool.
3. Enter the RCAHMS records into the database, again using the authoring tool.
4. Extend the language aspects to cover any missing nouns, verbs, and/or templates.

3.1.1 WP1 Tasks

The first task was to select a sub-part of the RCAHMS database with the aim of reflecting both the ease of translation into the M-PIRO format and the intrinsic interest of the objects.

The choice of data was made with the RCAHMS team at a workshop in the summer of 2004. The selection consists of 60–70 site records from Orkney and from Kinneil in West Lothian, along with associated text and image material. The records were chosen to include a range of site types, from Neolithic archaeology to 20th Century bridges and airfields; and also to be a fair representation of data quality, as some records have fully detailed information and others are quite sparse.

This task also involved the identification of which aspects of the RCAHMS data should be used, as the database has many more attributes than the sample museum database used by M-PIRO. The core fields from the main site table were used, and new fields were added to hold information extracted from the free text. Some information was taken from the RCAHMS table of archive material. In some cases the RCAHMS design had to be simplified for M-PIRO. The main reason for this was that, in its present form, M-PIRO does not permit multi-valued fields (ie one-to-many relationships between entities), so where these were needed a single value was picked arbitrarily, or else new entities were created to carry the attributes. For instance, the Barnhouse site is classified under “settlement” and “stone axes”, and here the “settlement” classification was kept, whilst a new “stone axe” entity was created under “finds” and related to the “Barnhouse” site entity. In other cases a secondary classification had to be simply dropped.

For the purposes of this pilot, the extraction from free text was done by hand, but the aim was to simulate what one could expect to achieve using automatic information extraction methods as explained in section 3.4 below. The core fields were used unchanged wherever possible (but see discussion in section 3.4), and were chosen either because they held key identifying information or because they would facilitate grouping and contrasting of sets of site entities. The complete list of attributes used for the pilot data structure is detailed in Appendix A.

Having selected the data, the next tasks were to create a type hierarchy following the example of the museum database and, when this is complete, to populate the database and add to the language entries where necessary.

Figure 1 illustrates the type hierarchy that was designed. The top level entities are: agent, classification, event, find, location, period and site. These are in turn subdivided into lower level entity types, and the actual data records come at the lowest level: the leaves of the tree. The structure of the site entity is shown in figure 2. The table at Appendix A also shows the breakdown of the entity types and the relationships between them. This is by no means a definitive design - the type hierarchy could have been constructed in many different ways. The design chosen was based around the particular subset of data being used, and reflects the kind of sites and buildings in that set.

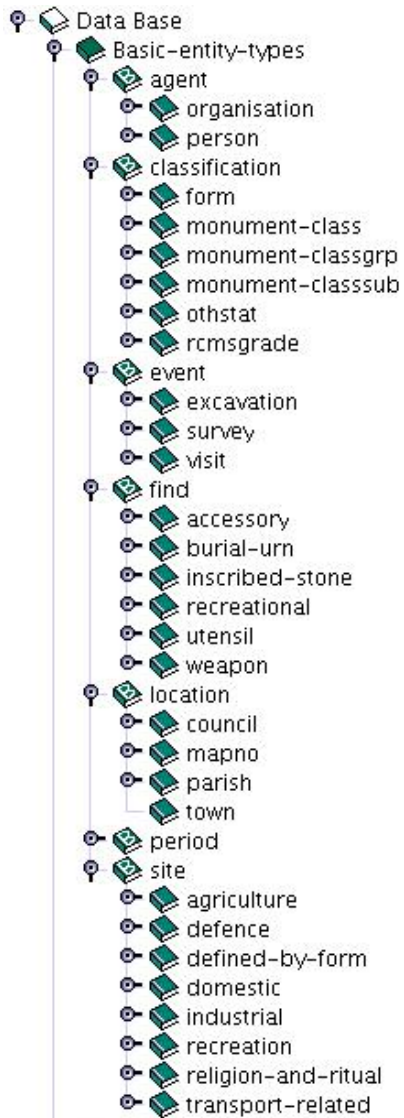


Figure 1: Type hierarchy

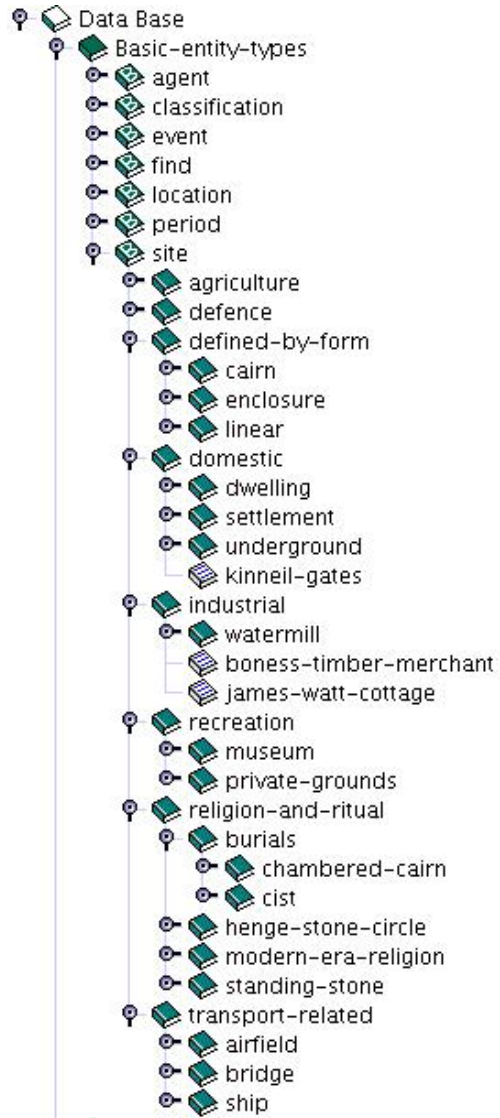


Figure 2: Site entity hierarchy

With the M-PIRO type hierarchy in place, the RCAHMS data could be entered, using the authoring tool. The most time-consuming part of the task then followed: adapting the language generation features to the data. This involves entering new nouns and verbs to describe data entities (such as “excavation”, “chambered cairn”, “classify” and so forth), and designing the clause plans and templates used to generate natural language sentences from the “facts” entered in the database. For example, given a site record with “Stenness” in the parish field, the application might produce a sentence like “This site is located in Stenness”. To avoid the text becoming too repetitive, alternative clause plans were provided, to be picked from at random, for the standard “adult” user.

A second user, “child”, was created, to demonstrate the ability of M-PIRO to generate simpler text from the same source data. Once the basic language model was built, it was possible to produce an adapted version using shorter sentences, fewer “facts” and simpler vocabulary. The order in which facts were presented to the user can also be varied, and the pilot illustrates this. For example, for the adult user the details of the site classification are given priority and the alternative site name is considered secondary; it will only appear on the first screen-full of text if there are very few more important facts to present. By contrast, the technical classification terms and their definitions are given lower priority for the child user; here “associated find” is one of the priority pieces of information. Changing the information ordering for different users is straightforward. The project highlighted one or two areas where there is scope for improving the way M-PIRO handles different vocabulary levels.

Although the domain has similarities to the Greek museum domain for which M-PIRO was designed, in practice there proved to be almost no overlap in vocabulary and terminology. (It seems possible that the dissimilarities have more to do with the difference between Greek and Scottish history and heritage, than with the change from museum to archive data.) However, once the language features have been put in place, they should be able to handle material from all parts of the RCAHMS database, not just the subset used here. The type hierarchy might need to be altered for new site types, but the changes to the language model would be relatively minor.

3.2 WP2: Front End Application

The main front end for the M-PIRO system was a web presentation, which is permanently maintained in Athens. Many aspects of the presentation were specific to the M-PIRO museum domain, and so it was necessary to write a new front end for the RCAHMS demonstrator.

The new web front end was produced using a client-server architecture. The server exposes the Exprimo functionality (such as creating users and getting user specific text describing artifacts). The client communicates with the server over a socket connection, and exchanges SOAP (Simple Object Access Protocol) messages with the server.

3.2.1 The Server Component

The server component consists of the existing Exprimo application with an additional wrapper that acts as an API into the Exprimo functionality. This component listens on a socket for SOAP messages containing commands and returns data in a SOAP message to the caller.

3.2.2 The Client Component

The client component is a web application running in a Java Server Page (JSP) container (we are using Tomcat). A single JavaBean (ExprimoBean) communicates with the server. Two other JavaBeans (Entity and Tree) are created to encapsulate and provide accessors to data on Entities and to the type/instance hierarchy. A set of Java Server Pages then provides the web user interface.

<i>Message Command</i>	<i>Command Parameters</i>	<i>Returned Data</i>
CREATE USER Creates a new user.	username The name of the user. usertype The type of the user, e.g. "adult"	SUCCESS or fault message
DELETE USER Deletes the specified user.	username The name of the user.	SUCCESS or fault message
ENTITY INFO Returns information on the specified entity. The returned descriptions are generated by Exprimo to suit the user, and take into account entities already seen and facts already conveyed.	username The name of the user. entityid The id of the entity. lang The language to use in the response.	text a plain text description of the entity. xmlinfo a marked up description of the entity with pointers to other information. title (optional) The title of the entity, if available. notes (optional) Notes for the entity, if available. image (optional) Name of entity image, if available. forwardpointers (not implemented) List of related entities. The Exprimo functionality this depends on is currently not working.
GET TYPES Returns an XML structure describing the type hierarchy for the domain.	lang The language to use in the response.	types An XML structure describing the type hierarchy for the domain.
GET INSTANCES Returns an XML structure describing the types of each entity in the domain.	lang The language to use in the response.	instances An XML structure describing the types of each entity in the domain.

Entity descriptions use the marked up xmlinfo data returned from Exprimo. This allows descriptions to include pointers to other entities or to additional information (such as descriptions of techniques or of places, when using the original M-PIRO data).

3.3 WP3: Assessing Scalability

The M-PIRO system was designed to work with a few hundred objects, and a number of questions arise when the task of scaling up to many thousands is undertaken.

Some issues have arisen with scaling up the current system, and we have identified several areas which will be improved in the forthcoming re-implementation. For example, the original system made assumptions about data structures which closely fitted the original museum exhibit domain, and we will ensure that the next version is more flexible in this regard.

Another problem is that once a user has seen all information on an entity, then further viewings give no information at all. The only way around this is to delete and recreate the user, which resets all data for the user. Ideally, the system would support either generating a final description for an entity that persisted for all subsequent viewings, or allow a user to start over for a single entity, as if they had not seen it previously.

During implementation of the web interface, it became apparent that some operations within Exprimo (e.g. generation of a description) expect there to be only one user of the system at a time. The server component works around this by only servicing one request at a time, but in a full-scale system this would be undesirable.

3.4 WP4: Investigating Information Extraction

As with WP3, this WP aimed at laying the groundwork for a larger project. The question here is how information extraction (IE) techniques might be applied to the RCAHMS database in order to create representations which can be used by Exprimo.

Information Extraction is the task of identifying key pieces of information in text and storing them in a format for further use: in this case, storing information in the M-PIRO type hierarchy. The task is generally divided into two subtasks, Named Entity Recognition (NER) and Relation Extraction (aka Template Filling).

NER is a process which identifies all strings in a document which refer to specific entities of pre-defined types, typically persons, organisations, locations, dates etc. In a separate project (SEER) we have experimented with RCAHMS data to investigate porting NER to a new domain and to extend our techniques to encompass entities which are terms rather than named entities. The RCAHMS domain has many standard named entities but other terms describing types of site, artifact etc. must also be reliably identified—the current M-PIRO application demonstrates this point clearly. We experimented with eight entity types, Organisation, Period, Person, Place, Reference, Site, Size, Date and in first experiments using general purpose machine learning techniques we achieved an overall coverage of 80%. Tuning of the system to the domain would yield higher accuracy.

Relation Extraction is the second process of IE which identifies relationships between the entities discovered by the first process. In the RCAHMS domain the relationships are typically between the site itself and the other entities, e.g. the period of site, artifacts found at the site, the architect of the building at the site etc.

For the purposes of this demonstrator, about half of the information used came directly from fixed data fields in the RCAHMS database, and wherever possible these were used unchanged. The rest was extracted from the free text in a way intended to simulate what would be possible with IE techniques as described above. Inevitably there were exceptions, and there was a strong temptation to improve the appearance of the demonstrator by tweaking data items manually.

Some examples:

- Site Description: The first two full sentences of the free text report were taken, which is something that could easily be automated. This works reasonably well in general, but in a few cases produces rather odd results and so manual alterations were made.
- Location Names: To make the text read properly, definite articles are needed before some council area names (eg “the Orkney Islands”) but not others (eg “West Lothian”). These alterations were made by hand. However, since the list of council areas is fixed and the changes have only to be made once, in the lookup list, this is a trivial matter.
- Classification Terms: Like the location names, these sometimes need articles in front of them (“a shell midden”) and sometimes not (“Roman pottery”). This is a slightly bigger job as the list of classification terms is extensive. Where terms are extracted from the free text it may be difficult to get the determiners right automatically.
- Thesaurus Scope Notes: These sometimes contain instructions to the archivist (eg “use specific term where known”) as well as term definitions aimed at the user. They were hand-edited as necessary, sometimes to create grammatical sentences, or to make them consistently singular rather than plural definitions (generally by prefixing the plural description with “a classification of...”).

We would expect an IE system for the RCAHMS domain to achieve fairly high levels of accuracy but we do not foresee that IE would be used to fully automate the process of transferring content from the RCAHMS database to the M-PIRO type hierarchy. Instead we suggest that IE can be deployed as a tool to assist human curators, probably using a version of the current M-PIRO authoring tool as an interface.

A Data attributes used in demonstrator

M-PIRO field	RCAHMS source	Remarks
site.name site.mapno site.parish site.council site.classgrp site.classn site.classsub site.rcmsgrade site.form site.othstat site.period site.altname site.associated-find site.visit site.survey site.excavation site.description	rmain.nmrsname rmain.mapno rmain.parish rmain.council rmain.classgrp rmain.class rmain.classsub rmain.rcmsgrade rmain.form rmain.othstat rmain.altname rctextrep.report rctextrep.report rctextrep.report rctextrep.report rctextrep.report	Pointer to location.mapno.name Pointer to location.parish.name Pointer to location.council.name Pointer to classification.monument-classgrp.name Pointer to classification.monument-class.name Pointer to classification.monument-classsub.name Pointer to classification.rcmsgrade.name Pointer to classification.form.name Pointer to classification.othstat.name Pointer to period.name Pointer to find.name Pointer to event.visit.name Pointer to event.survey.name Pointer to event.excavation.name First two complete sentences of free text
agent.organisation.name agent.person.name	various various	Organisation name associated with site visit, survey or excavation. Extracted from rctextrep.report or from rcollect.desc1 or rcollect.surname. As above, but personal name
classification.scope-note classification.form.name classification.monument-class.name classification.monument-classgrp.name classification.monument-classsub.name classification.othstat.name classification.rcmsgrade.name	thesaurus scope note	Distinct rmain.form values Distinct rmain.class values Distinct rmain.classgrp values Distinct rmain.classsub values Distinct rmain.othstat values Distinct rmain.rcmsgrade values
event.event-site event.person event.organisation event.excavation.name event.survey.name event.visit.name	 rctextrep.report rctextrep.report rctextrep.report	Pointer to site.name Pointer to agent.person.name Pointer to agent.organisation.name Date of main excavation. Has to be extracted manually from free text. Date of last survey. Taken from free text, looking for strings around keywords such as “resurveyed”. Date of last visit. Generally looking for last occurrence of “Visited by...” strings within free text, so could be found automatically with fair accuracy.
find.site-found find.classification		Pointer to site.name Pointer to classification.monument-classsub

find.period find.date find.*.name		Pointer to period.name Mostly provided for each find individually by RCAHMS staff, or extracted from free text Based on rremain.classsub or extracted from free text, for each of the six categories of find used.
location.council.name location.mapno.name location.parish.name	rccouncil.couname rremain.parish	Distinct rremain.mapno values
period.name		Mostly provided for each site or object individually by RCAHMS staff; present in rremain.period in some cases