The Scavenger Approach to Data Reclamation and Acquisition from Mixed Media Original Sources

Abstract

This paper describes the Scavenger approach for the reclamation of data from unstructured "mixed media" original sources. This approach provides semi-automated tool support to assist human operators in populating highly flexible data models using information derived from large sets of pre-existing source materials. The data models generated by this process are stored in such a way that they become accessible for manual investigation, browsing and importation into existing tools and processes. Scavenger is a generic approach which can be applied to many different domains. The approach addresses the difficult problems associated with the structuring and manipulation of large sets of unstructured original source material, derived from potentially many different sources.

Keywords: Knowledge and Data Engineering Tools and Techniques (E.0.e)

Document Capture – Document Indexing (I.7.5.b)

System Applications and Experience (E.0.f)

1 Introduction

Many modern work activities produce large amounts of complex information laden material as their output. Ethnography is no exception, producing much written text, original artifacts, photographs and audio records which form part of the description of a particular domain. The sheer size and extend of this information makes it both difficult to manage as well as manipulate. In addition to this, the plain format and lack of extensive internal structure present in these

materials makes both manual and automated processing difficult. This is typical of the output of many human centred activities which attempt to construct in-depth documentational or descriptive representations.

Attempts to insert additional structure into such information are hampered by the labour intensive nature of the work and the lack of suitable support tools. As we shall see later, activity specific tools do exist but they are usually not generic enough to be applied to a range of different activities and their specific outputs. This results from the fact that each tool is based around various domain specific concepts and constructs which are ingrained within the operational processing of that tool. For these reasons, we have developed the Scavenger toolset.

Scavenger is a simple but powerful approach which supports the direct recovery or "reclamation" of data from ethnographic reports and original source materials. We do not utilise fully automated grammatical analysis techniques due to issues relating to accuracy, scaleability and generalisability of such approaches. Rather we prefer to keep the human "in the loop" by providing tools to enhance and augment the activities of a human analyst. To support this process, the tool provides mechanisms for the selection of fragments of media from original source documents and the population of the structured models using a specialised "cut and paste" mechanism.

In this paper, we shall first discuss some of existing approaches which fall short of providing the facilities supported by Scavenger. We will then describe various key concepts from data modelling which are relevant to the process of scavenging. Following this, we will present the data storage and data reclamation mechanisms utilised by the Scavenger toolset. Finally we shall discuss the suitability and efficiency of the toolset by examining two different approaches which

utilise Scavenger as a mechanism for data reclamation. With these utilisation case studies, we hope to demonstrate the utility of the Scavenger approach.

In all of this discussion, we should bear in mind that although Scavenger was initially developed for use with the output of ethnographic activities, it is equally applicable to supporting any task which involves the generation and manipulation of large quantities of minimally structured, media rich information. Scavenger may be used to explore, capture, structure and collate fragments of essential information from any domain in which relatively unstructured and natural language documents and collections of pictorial, photographic and audio evidence exists. We are however particular interested in the manipulation of the results from ethnographic activities as these posses a number of challenging problems which may be suitably addressed using the Scavenger approach.

2 Background

In this section we provide a description of the background context which has influenced the development of the Scavenger toolset. This includes the aims and motivations behind the Scavenger concept as well as the influence and deficiencies of existing work in this area. In so doing, we hope to provide an illustration of the need for and intended target domain of the approach. We also aim to define the objectives and applicability of this work, both of which will be essential for assessing the success of the Scavenger approach.

2.1 Aims of scavenger

The main aim of the Scavenger approach is to support the investigation, selection, structuring, sorting, composing and organising of fragments of data from original source materials. Such

original sources include electronic manuals, paper based manuals, photos, handwritten notes, paper artifacts, recorded audio and so on. Scavenger is particularly generic and may be utilised in a variety of domains providing that pre-existing original sources in either electronic or physical format are available. It is however important to note that all physical sources must be digitised in some way so that they may be manipulated by Scavenger.

Practically, the Scavenger approach attempts to ease the process of creating and populating structured models by providing partially automated data reclamation facilities. These are only partially automated since they require tool user participation and are not based solely on automatic extraction. On one level, Scavenger can be thought of as a sophisticated, yet speedy and efficient "cut and paste" mechanism which is tailored towards the population of a modelling database. Scavenger can be thought of as a "funnel" to ease the process of decanting source material into a structured modelling framework. It is important that when performing this role Scavenger should support, but not constrain analysis or the model construction processes. Thus it is essential to provide flexible, useful and appropriate mechanisms to support work practices.

Scavenger provides a variety of generic data acquisition mechanisms for various types of media to support the recovery of fragments of data from existing artifacts. These mechanisms are constructed in such as way that they may support modelling for a variety of different applications. The output of Scavenger is an open, structured database which may be subsequently accessed for further interrogation and processing. A key feature of Scavenger is that it is model independent and can be used in the construction of a wide variety of entity-relations based models. So that this database may be of most utility, it's format and structure is specified by the "customer" application which is to make use of the data after scavenging has

taken place. Subsequent usages of the data models generated by Scavenger could involve browsing by humans, advanced querying, data re-factoring, automated analysis and so on.

By allowing users of the Scavenger tool to specify the database schema before any data reclamation takes place, it is also possible to ensure that the process of data collection is performed according to a standard format. This provides both consistency and conformity to simplify and support the activities of individual users as well as across groups of users working in a team. This regular format also makes the process of browsing and investigation simpler and more consistent after scavenging has taken place. In addition to this, the uniformity of the pre-defined schema also simplified browsing by humans and importation by existing tools.

In terms of our main area of interest, that of the manipulation and structuring of ethnographic information, Scavenger can offer a number of concrete benefits. We believe that the suitable structuring and presentation of ethnographic information is essential to support both investigation and comprehension by system designers, engineers and other interested parties. In addition to this, much can be gained from the pre-processing of raw ethnographic data in order to distil, refine and restructure it into an appropriate format to support direct access by software tools and methods.

2.2 Existing approaches

In this section we highlight some of the key technologies and approaches which may be used to construct structures of the type described in the previous section. At a basic level, some of the generic features included in most operating system provide support for data recovery and

structuring. The cut and paste facilities found in modern operating systems is one such feature which support the recombination of original source materials into a more structured forms.

Cut and paste mechanisms are further extended by technologies such as Object linking and embedding (OLE) [1] present in recent versions of Microsoft Windows. OLE permits the embedding of media created by a variety of different applications inside a single application (typically a word processor). Such media can include text, formulae, graphics, images, structured diagrams, audio, video and so on. This mechanism not only allows viewing of media, but interaction and editing functionality as well. Similar technologies include most notably OpenDoc [2] for OS/2, Mac OS and Unix/Linux platforms, as well as proprietary mechanism commonly found within office application suites.

Although such low level mechanisms provide basic and commonly used mechanisms for structuring original source materials, much can be gained from using richer, more sophisticated mechanisms specifically tailored to this task. There are a wide variety of manual, semi and fully automated approaches which attempt to address this problem. Fully automated approaches include those based on natural language processing, wrapper induction and ontology generation [3]. These however are not applicable due to our need to support media of a variety of different types and the lack of internal structure which they exhibit. In addition to this, the augmentation of such media is a highly creative task requiring the skill and domain knowledge of a human operative. Existing automated approaches are not yet advanced enough to provide a suitable solution.

Manual support and semi-automated approaches such as NoDoSE [4] do however provide the opportunity to enhance the process of structuring by a human operator. NoDoSE is a tool for

extracting structured data from textual documents such as e-mail files, electronic documentation, OCRed printed material and so on. NoDoSE utilises "clues" given by a user as to interesting areas of semi-structured data. The user utilises a graphical interface to highlight and identify the different elements of a particular file. In this way the user can construct a hierarchical structure for a document, consisting of a number of nodes and sub nodes, each of which may be associated with a number of name-value pairs (known as fields). Using information from this manual selection phase, NoDoSE is able to extract structures from all similar documents. If mistakes are made in this reclamation process, the user is able to correct the automatic extraction. The new clues indicated by such corrections can then be utilised by the tool in order to incrementally improve the automatic extraction process. Once structures have been extracted from documents, NoDoSE can make these available in a number of standard formats including comma delimited files and external databases for importation by DBMSs.

The most common tools currently in use in ethnography for structuring the generated documents are qualitative data analysis approaches [5], [6], [7], [8]. These tools utilise the mechanism of "coding" to allow users to associate keyword handles to fragments of ethnographic material. Segments of media are interactively selected by a user and then tagged with existing or new codes. These codes allow ethnographers to insert additional structure into ethnographic material in a very free form manner. Using these codes, it is then possible to extract related sections of a diverse set of documents, as well as performing complex queries to derive useful information regarding the system under investigation. This includes the ability to identify related sections linked by the same codes as well as the intersection and overlapping of codes relating to similar areas or concepts. Of particular interest amongst qualitative analysis approaches is Atlas/ti, which supports a wide range of media types, ranging from electronic texts, through graphical

images, to basic support for recorded audio. Atlas/ti allows users to specify basic relationships between codes using families (parent-child relationships) and networks (general associations).

The table below summarises the approaches mentioned in this section and indicates the extent to which they meet the aims and objectives mentioned earlier in this paper. We include the new Scavenger approach for comparison purposes.

Table 1 Comparison of approaches

	Unstructured data source	Support for selection	Support for importation	Support for mixed media	Consistent Schema	User specified Schema	Open access database
OLE/OpenDoc	✓	✓	✗	✓	✗	✗	✗
NoDoSE	✗	✓	✓	✓	✓	✗	✓
Ethnograph	✓	✓	✓	✗	✗	✓	✗
Atlas/ti	✓	✓	✓	✓	✗	✓	✗
Scavenger	✓	✓	✓	✓	✓	✓	✓

2.3 Problems with existing approaches

Despite the existence of a number of approaches for the structuring of data from original sources, a number of issues still remain unresolved. One of the immediately obvious problems with existing approaches are the constraints which they place upon data analysis and restructuring tasks. This ranges from rigid data collection processes though to the simple lack of support for various important types of media. These have the effect of constraining the behaviour and inadequately supporting the natural work of the tool users. Some automated approaches also undertake too much control, thus excluding human intervention in the data collection process. This can prevent the use of important human knowledge, skills and abilities to the gathering and structuring process.

A number of approaches require some structure to be present in a document in order to perform extraction. Ethnographic documents are usually textual and verbose and often do not have adequate structure to permit such extraction. This reduces the effectiveness of approaches such as NoDoSE, making extraction inaccurate, over conservative and error prone. In addition to this, such partially automated approaches cannot currently operate efficiently on non-textual original sources due to the need to be able to analyse the content and structure of material. This makes such approaches impossible with images and at best unreliable with hand written notes.

There are many problems with existing approaches in regard to the content and makeup of structural models which they generate. Most quantitative analysis approaches produce structures with very course grained elements. This results from the fact that whole sections tend to be tagged with single codes. In such situations, extensive or rich structures are hard to achieve. This can be especially confusing when combined with the fact that such codes act as a multi-purpose flags and the exact reason for their use is not clear. Due to the flexibility and unconstrained nature of codes they liable to value judgements and thus are often be personal to the ethnographer who identified them.

Additional problems result from the way in which relationships are created in such approaches. Relationships between codes are liable for omission since it is up to the user to create them and there can be no automated checking because the tool has no concept of the semantics of a particular code. The way in which different codes relate to each other is not always clear nor captured by such approaches. We must also consider the situation where a newly created code is relevant to sections previously investigated, but not tagged with that code since it did not exist

when those section were covered. Such a situation may require an ethnographer to go back over previous sections to check to see if they require tagging with the new code.

Many approaches are tied to a particular domain, mainly due to the fact that the support tools needs to have knowledge about the entities which are to be stored. As a result of this, such approaches are often hardwired with fixed data models that cannot be altered or adapted to meet the varied needs of the users. This does not provide users with the flexibility to model what they wish, in the manner that they wish, leading to a serious constraint upon their work practices.

Despite the desire to free the operator from the constraints of a fixed structural model, there is also the potentially conflicting need for a regular and standardised schema to support understanding and investigation by all other parties. This is particular important to aid investigation by domain novices who will benefit from a predefined and regular structure. Pre-defined and standardised entities and structure can also be beneficial to data extraction, model checking and other similar processes. It is thus important that we find a balance between proving a consistent structure for the modelling database, without constraining the modelling or work of the users.

A final criticism which can be levelled at the existing approaches described in the previous section is the lack of provision of an open database format to support access by existing or future third party tools. As a result of this, access to and investigation of data is limited to tool support provided with the approach. Users must often be content with the limited speculative browsing, data mining and visualisation facilities offered by a particular approach. Occasionally, limited support is provided in the form of export to SPSS compatible files or basic comma/tab/space

delimited flat file formats. This is however a poor replacement for access to a rich, deep and fully structured open database holding all structural modelling information.

3 Scavenger database

In this section we describe the structuring and storage mechanisms used by Scavenger in an attempt to address many of the issues raised in the previous section. We also detail the way in which users of the Scavenger toolset may specify a schema for the stored data so that it is most useful to then for the purposes of modelling and later interrogation by other tools and techniques.

3.1 Data model

The data model used by Scavenger is based on the entity-relational paradigm which is common to in many database and data modelling mechanisms. We have taken such concepts and have adapted them for use within the Scavenger approach. The primary element of the Scavenger data model is thus the "Entity". An entity is a generic element found in many data modelling approaches which can be used to represent a wide variety of different phenomenon. An entity may represent a person, a piece of software, a management structure, a physical location, a domain artifact and so on. An entity is a concept which is similar in some ways to a "code" from the previously mentioned qualitative analysis approaches. An entity does however have a number of key differences. Firstly, entities are limited to concrete and observable phenomenon from within a domain whereas codes can represent a much wider range of concepts ranging from phenomenon and processes to thoughts, theories and observations. In addition to this, entities have standard and definite relationships with other entities in a system. Codes also tend to be

courser gained "summaries", were a single code may be equivalent to a number of different entities combined together.

Within the Scavenger approach, each entity which is identified consists of the following elements:

- Class - representing the type of entity
- Name - textual or graphical identifier for the benefit of human users
- Description - textual or graphical explanation of the entity for the benefit of human users
- Flags - boolean values indicating the nature of the entity
- Relationships - links with other existing entities

If such entities form the basic building blocks of any Scavenger derived representation then the relationships between them form the cement which holds the model together. Relationships represent logical associations between various entities and can record of a wide variety of information. Once an entity has been created, relationships between that entity and any others may be created. The notion of such relationships are again a standard concept from entity-relational modelling. In Scavenger each relationship is always associated with an inverse relationship. This permits bi-directional tracing between the entities linked by a particular relationship.

Each class of entity within Scavenger is associated with a fixed set of relationship types. Any number of relationships of each type may be associated with an entity, whereas the set of relationship types associated with an entity class cannot be changed from those originally specified before Scavenging commences. This contrasts with the concept of "codes" used in

qualitative analysis, the relationships between which are optional, non-predefined, ad-hoc and liable to omission.

To help illustrate the use of relationships, let us consider a hypothetical "person" entity. Each entity of this type has the following relationships associated with it:

- Parent - a relationship to one or more parents, each of which will also be a "person" entity.
- Employer - a relationship to one or more companies for which a person works, each of which will be a "company" entity.
- Home - a relationship to one or more addresses to which the person lives, each of which will be an "address" entity

3.2 Schema specification

As we have already explained, the data model used by Scavenger is one based on the entity-relational paradigm. The database schema which is used for a particular data model is totally open and may be configured to meet the data requirements of the tool user. The schema for the E-R model is defined using a set of XML templates which may be created by any user. This means that Scavenger can be use in the construction of any E-R based models by simply "plugging" a new set of XML templates. These XML templates allow a user to specify the set of possible entity classes. Each entity will have a textual or graphical name label, a description, zero or more flags to indicate the nature of each entity and a set of relationship types between entity classes. The relationship lists are the most complex of all constructs and detail the name of the

relationship, the entity type of the target of the relationship and the name of the inverse relationship.

Entities are created using such templates as an initial outline and then populated using data scavenged from original sources. Because users are able to specify the exact nature of entities and relationships, the output database of the scavenging process may be tailor to meet the needs of that user. This can allow the shaping of the data into a form that can be manually investigated, browsed, imported into existing tools, translated into other formats, visualised, analysed or explored using data mining techniques.

3.3 Data storage

The previous section described the way in which the database schema is specified, we shall next move on to explore the mechanism used to store the actual entities and relationships which have been created in compliance with this schema. Each entity which is created is constructed according to XML template of the class to which it belongs. This will initialise the flags and relationship lists for the entity instance which is created. Once created, each entity is stored in a separate XML document and the relationships which link it to other entities are mapped to document links specified using special XML tags. These XML documents are then stored directly on the file system to ensure ease of access by the user and other applications.

XML documents were chosen as the storage mechanism for Scavenger data because such documents provide an open data storage mechanism. Due to this open nature, entities are human readable and can be viewed, manually edited and browsed by an operator. To help support this, Scavenger utilises XSL style-sheets to render browseable HTML pages for each entity, based on

the XML files which have been generated. The XML documents are dynamically translated into HTML and can then be navigated using any modern web browser. In addition to this, the open XML document structure also allows the generated database to be accessed by existing or future tools, methods and approaches. This final feature is greatly assisted by the wide availability of XML parsers for use by developers in the creation of such tools.

4 Data reclamation

Now that we have some idea of the structure and storage mechanisms used to handle collected data, we may go on to address the methods used to populate this database. Data reclamation is the term given to the human driven extraction of key fragments of information from the original sources and the importation of these pieces of data into the structured model provided by Scavenger. In this section we will discuss all aspects of this data reclamation, including the tools provided to browse and select segments of source media, the mechanisms used to populate entities with selected media and the construction of links between the created entities.

4.1 Media fragment selection

Scavenger supports a wide variety of media types in order to maximise the number and diversity of original sources which may be used to populate the Scavenger database. Media selectors allow Scavenger users to browse source materials and select suitable fragments for insertion into the structured models. We have developed a variety of different types of selectors to support a wide range of source media. By providing a variety of such selectors, it becomes possible to support the reclamation of data from all of the main types of ethnographic source material which are available. The media selectors which form part of the Scavenger tool are as follows:

- Textual - Allows an operator to select strings from existing plain text, html or rtf documents. (e.g. electronic reports, audio transcripts, electronic manuals etc.). Utilises text highlighting to indicate areas of current and previous selection. All instances of a selected word or phrase are selected and we are currently working on stemming algorithm to extend this. Fragments are grabbed as plain text into Scavenger.

- Graphical - Allows an operator to select fragments of images (e.g. images of handwritten notes, photos, scanned artifacts, diagrams etc.). Utilises fragment highlighting to indicate areas of current and previous selection. Fragments are grabbed as images into Scavenger.

- Audio - Allows an operator to select short audio clips (e.g. interviews, telephone conversations, audio diaries etc). Utilises section highlighting on audio timeline to indicate areas of current and previous selection. Hyperlinks to selected clips are imported into Scavenger.

- Application - Allows an operator to select fragments of text from other applications (e.g. word, powerpoint, web browsers etc). Does not allow text highlighting to indicate any areas of current or previous selection. Fragments are grabbed as plain text into Scavenger.

Figure 1 below illustrates a graphical media selector being used to scavenge handwritten notes from a scanned notebook. The dark highlighted areas indicate previously selected fragments which have already been imported as data for use in defining entities. The lighter areas indicated a selected fragment which has yet to be imported.

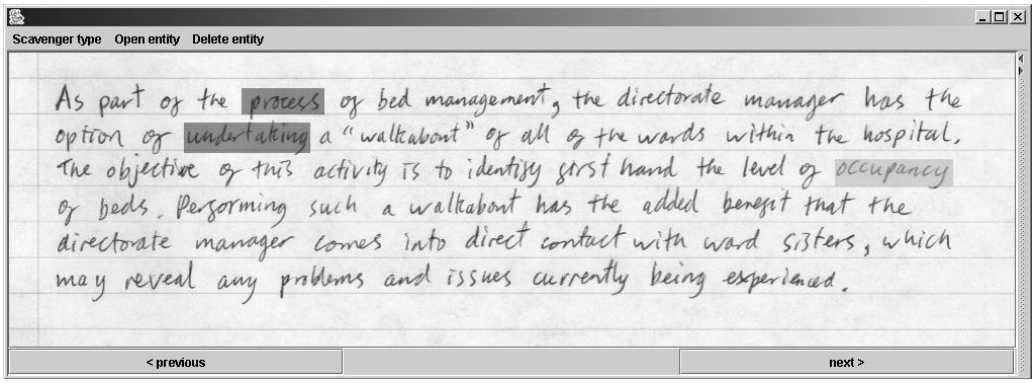

Fig. 1 Graphical media selector

Scavenger does not concern itself with the meaning of entity names and descriptions, since it performs no operations based on the semantics of these items. Only the human operator needs to be able to observe and understand such data items. For this reason, images of handwritten text may be utilised as though it were electronic texts. These are purely labels presented for the benefit of the human operator. Such graphics are presented to the user in an identical way to normal textual data labels (e.g. in "text areas" and "selection lists")

4.2 Entity creation

The structuring of data derived from original source materials takes place in three different modes. These are entity creation, entity population and entity completion. Although we describe each of these separately, the boundaries between them can blur and the activities of each overlap. The first of these, entity creation, involves the tool user rapidly and dynamically create new entities for storage in the Scavenger database. Entities are created using one or more selected media fragments which form the initial name for that entity. The user simply has to select a single or sequence of fragments and then instruct the tool to construct a new entity based upon them. At this juncture, the tool user specifies the desired entity class which is required and a new

instance of that class is created using the appropriate entity template as an outline. This creation process is activated using the entity creation menu provided by the Scavenger tool. The use of this menu is illustrated in figure 2 below.

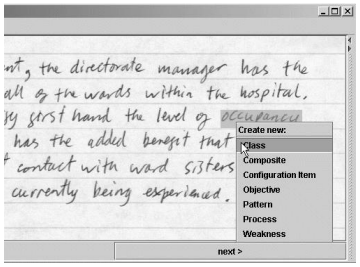

Fig. 2 Entity creation menu

Using this model of data reclamation, Scavenger allows tool user to rapidly create new entities. The speed of this operation and low creation overhead is essential to support the work of the tool user. Decisions to create a Scavenger entity to model a concept or construct from the study domain may occur in rapid succession and in large numbers to the tool user during browsing and investigation of the original sources. It is most important to support and not constrain these thought processes and activities of the tool user or analyst.

Scavenger automatically keeps a record within each entity of all the original source documents from which fragments have been scavenged in order to construct the entity. This provides a source traceability mechanism which allows an operator to trace back from a particular entity to all the original sources which have contributed to that entity. This traceability is fairly coarse grained and is implemented at the document level. This means that Scavenger will indicate the source file (e.g. scanned page, electronic document, audio file etc) from which an entity was derived and it is up to the operator to identify the exact point in that document which contains the actual scavenged media fragment.

4.3 Entity population

Once created, the empty XML document representing the entity is ready for further additions either by manual editing or population using the second mode of scavenging. This mode is known as entity population and allows the tool user to gather together fragments of original source media and quickly and efficiently insert them into the stored entity of their choice. Both the name and the description of an entity can be populated with fragments of scavenged media in this way. Within Scavenger, an entity editor is used by the tool user to indicate the elements of an entity model which is to be filled by the imported fragments of ethnographic source media. This is illustrated below in figure 3 which depicts the use of a textual media selector to populate the description of the created entities (using the last sentence from the original source on the left hand side)

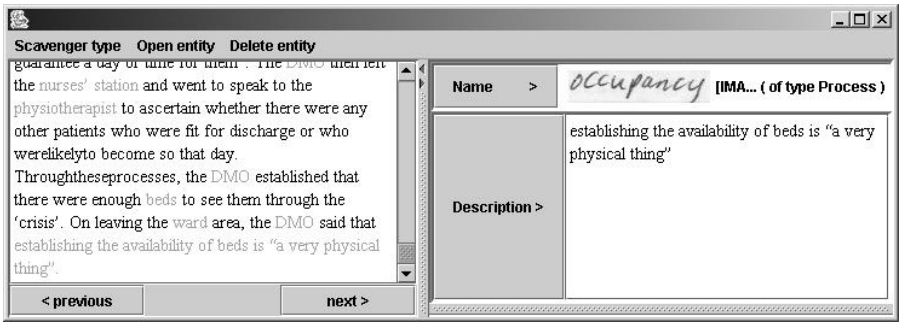

Fig. 3 Entity editor

Scavenger facilitates the rapid and efficient population of database entities with the above features. This process must be fast and efficient due to the often significant number of entities which exist within a database, the large quantity of original source material and the high throughput which is required. Any number of fragments may be imported to provide name and description materials for a particular entity. It is important to note that there may be many

different cycles of population and an operator may return any number of times to update, alter or replace the name and descriptions of each entity. As with the creation of entities, the source document of all scavenged media fragments used in the population of entities are recorded by Scavenger. This allows backwards traceability from each entity to all of the source materials which have contributed to its construction.

5 Entity completion

Data reclamation can only be expected to partially complete the created Scavenger entities. This is because not all data may be explicitly contained within the original source materials. Some important information may be implicit or implied and therefore not available for selection by the media selectors. Other data may not be evident at all from the information held in the source materials. Certain parts of entity descriptions as well as the relationships and flag values associated with them will have to be added manually by the tool operator.

In such a situation, considerable emphasis is place on the abilities of the tool user or analysis to identify and manually enter the relevant information. For these reasons it is not possible to fully automate the process of Scavenging and the human operator must remain "in the loop". The best we can hope to achieve is to support the activities of the tool user as efficiently as possible in order to make the identification and entry of this information as fast and as accurate as possible. Using the XML templates, Scavenger can dynamic create the correct entity editors which will support the entry of the required information. This is illustrated in figure 4 below where the flag checkboxes (including checkbox groups) and relationships lists are dynamically created based on the template of the class to which the entity belongs.

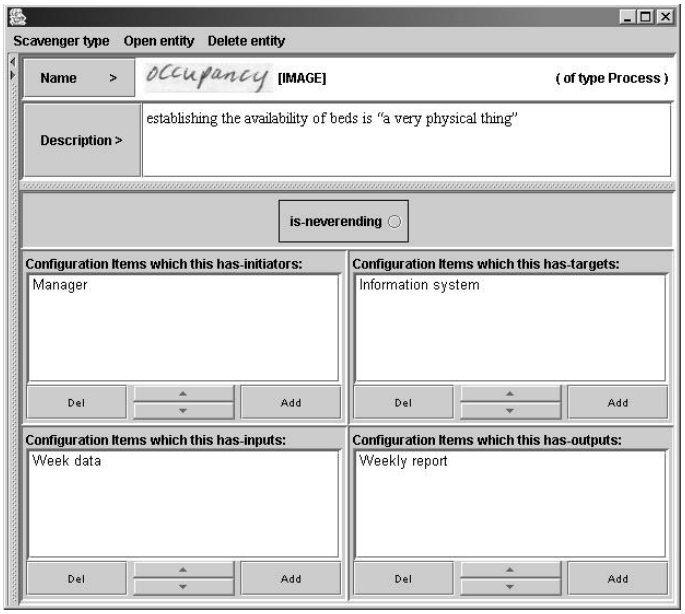

Fig. 4 Custom entity editor

Using entity editors such as the one above, it is possible to add to or alter the information held about an entity at any time. Flags are altered by simply clicking on the desired option. Relationships are created by adding existing entities to the appropriate relationship list. The inverse relationship for each forward relationship is automatically created by Scavenger. For example, if entity B is added to the "is-parent-of" relationship list of entity A, then the entity A is added to the "is-child-of" relationship list of B.

6 Data utilisation

Scavenger's open and independent data storage mechanisms allow reclaimed data to be access by any number of different "customer" processes, tools and approaches. Scavenger can be though of as a pre-processor of raw original source material in order to make it suitable for utilisation by subsequent approaches. By allowing the tool user to specify the schema used for the structure of the Scavenger database it is possible to ensure that whichever activities follow scavenging will

be provided with a rich, structured data source that is tailored to their specific needs. Each activity will have it's own requirements and constraints upon the data which it utilises. By taking advantage of the XML template based schema specification facilities provided by Scavenger, such factors can be taken into account and incorporated into the database produced.

The types of activity which may be performed on the Scavenger database after creation are wide and varied. At a basic level the produced database may be direct viewed and edited manually by a user, perhaps with the aid of a text editor, word processor or spreadsheet application. Although possible, manual investigation of this form is slow and difficult. For this reason Scavenger provides support for the specification of XSL style sheets which can be used to translate the XML documents into HTML pages which in turn can be browsed more effectively using any modern web browser.

In addition to this basic form of browsing, existing or specially developed tools may also make use of the produced Scavenger database. These may include tools such as data mining applications, automated analysers and reporters, visualisation tools and so on. Such tools can either convert or import the Scavenger XML documents and store them in an internal format, or alternatively operate directly on the XML documents themselves.

7 Evaluation

In order to evaluate and assess the utility and applicability of the Scavenger approach, we will now document the successful adoption of Scavenger mechanisms by two separate modelling approaches. In so doing we will describe how the flexible schema specification mechanisms are used to create suitable databases for access by the two approaches. In this section we will attempt

to identify the properties of Scavenger which make it so suitable for the adoption as a pre-processor to such approaches. In addition to this, we will also highlight some of the lessons learnt and insights gained during the process of adoption and integration.

7.1 Usage study 1 - Situation Modelling

The first usage study focuses on the adoption of the Scavenger approach for data reclamation in Situation Modelling [9] Situation modelling can be thought of as the sophisticated modelling of environments and provides a structured ethnographic description of a particular work situation. The aim is to provide a way of presenting situation information to designers in order that they can gain a better understanding of the system requirements and various domain issues. In this way situation modelling attempts to inform the design process and improve the quality of the finally produced system. Situation modelling allows designers to get first hand experience of the context in which their system will be deployed. It also provides a way for ethnographers to communicate their findings to a wider audience and at the same time provide them with a tool that will aid in the compilation of reports.

A situation model should provide an overall view of the situation from the basic hierarchy of elements, to very detailed information about specific tasks. This is a great deal of information to accommodate in one model and so the notion of perspectives or viewpoints are needed. There are three main viewpoints, the Task viewpoint which looks at large granularity work tasks, the Role viewpoint which looks at operational roles within an organisation, and the Artifact viewpoint which looks at artifacts from the application domain. The entities which form these viewpoints are listed below, some are shared between viewpoints and others are unique to a particular viewpoint:

- Stream - A stream is an ongoing series of activities that has some purpose, with no clear beginning or end.
- Activity - An activity is a set of defined/recognisable units of work with a specific purpose. It is a unit that has a beginning and an end, but can be made up of sub activities. Activities are also made up of events and may be ordered into coherent sets called Event sets. These may be carried out one or more times.
- Event Set - An Event set contains smaller events that must be carried out in order to complete the activity. The events may need to be carried out in a specific order.
- Sequence - An ordering or partial ordering of events or activities.
- Event - These are units of work that can be observed/identified and have an observable beginning and an end. They are atomic.
- Explanation - This is an explanation of the event from a certain point of view. There may be more than one explanation for a single event.
- Vignette - These are short examples, or instances of the events, they can be used to further explain the event, or a viewpoint.
- Placement - The placement defines where the role is within the hierarchy of the situation.
- Interaction - Interactions describe the actors and artefacts that this particular role encounters.
- Starting Environment - The starting environment describes where the artefact begins, where it starts off life.
- Setting of use - Where the artefact is required and by whom. This highlights the interactions between the roles, actors and various tasks.

Scavenger can be used to construct entities of all of the above types by performing reclamation of data from ethnographic original sources such as photographs, domain artifacts, audio records, handwritten fieldwork notes, electronic reports etc. In this role Scavenger is basically being used to add additional form and structure to the original ethnographic information. By specifying an XML template for each of entity type, a Scavenger database schema can be created which conforms to the data model mandated by the Situation Modelling approach.

The entity classes specified using such templates and the subsequent entity instances created and populated by Scavenger form the core to the database which is produced by data reclamation. Situation modelling can then utilise this database to help inform the design decisions of system developers, managers and so on. To achieve this, it relies on the translation of XML documents to HTML web pages, which can then be browsed by interested parties involved in system development. An example of one such browseable HTML page is shown below in figure 5.

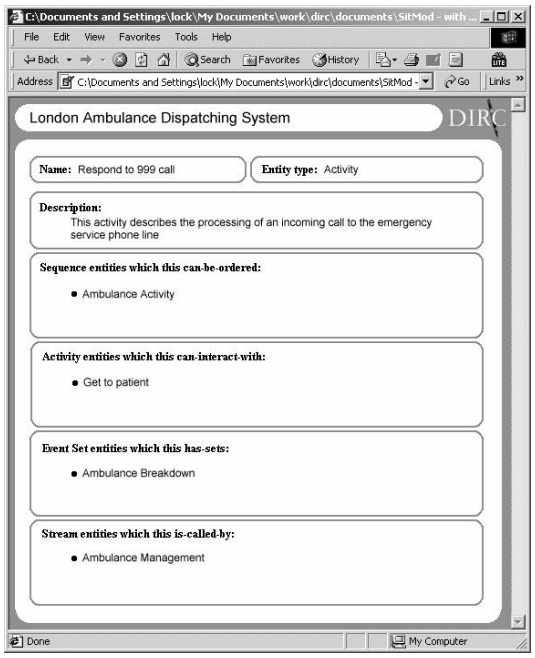

Fig. 5 Example rendered entity

Scavenger handles all creation, management and alteration of the entire structured models upon which Situation Modelling is based. The activities which are carried out upon the Scavenger database after scavenging has taken place are all investigation, explanation and discussion oriented tasks performed by human operatives. Although no automated activities are performed upon the Situation Models produced by Scavenger, a support tool is planned which will include a variety of querying and reporting mechanisms which will operate upon this data.

7.2 Usage study 2 - Configuration modelling

The second usage study is based on the adoption of Scavenger for data reclamation for Strider configuration models [10]. The aim of Strider is to support the modelling and analysis of the top-level configuration of socio-technical systems. The focus of Strider is on the composition and structure of a system's configuration, rather than the operation of the system itself. The main aim is the modelling and dependability analysis of the configuration which currently pervades in an existing socio-technical system. Strider also allows users to formulate new proposed configurations, based on concrete knowledge about existing components and analyse them to assess the dependability of that intended configuration. Automated analysis features provided by Strider and access to the configuration models themselves allow system stakeholder to achieve a better understanding of the system and gain insights into the dependability attributes of current and further configurations.

In the Strider approach, configurations consist of components and the processes which related those components together. Components can represent a wide variety of phenomenon found in socio-technical systems, ranging from pieces of software and hardware, through buildings and physical spaces, to the people who work in an organisation. The Strider models are based around

two separate perspectives of a system, structural model (demonstrating "part-of" relationships) and a set of process models (demonstrating abstract "used-by" relationships). The Strider entities which Scavenger can be used to construct include the following:

- Configuration Item - A component which forms part of a system configuration
- Class - Defines a particular class of configuration item which may be present in a system configuration
- Process - A work process which relates a number of configuration items together
- Composite - A commonly experience, complex configuration "cluster" (focusing on structure) which is constructed on multiple atomic configuration items
- Pattern - An abstract commonly experience, complex configuration "cluster" (focussing on process) which involves multiple atomic configuration items
- Objective - A high level task or aim the attainment of which is supported by the system configuration
- Weakness - An identified problem or anomaly associated with a process which may prevent the achievement of objectives

The role of Scavenger within the overall Strider approach is to identify, describe and structure key system components which are derived from an input set of ethnographic source materials. Scavenger also permits relationships to be specified between entities (e.g. sub-class, instance-of, part-of, has-input etc). In order to achieve the Scavenging of components and their formatting into an appropriate data structure, an XML template has been specified for each of the previously mentioned entities.

The Strider tool operates directly on the produced Scavenger database, utilising a standard XML parser to obtain access to the data held within. Based on this data, Strider is able to produce graphical representations of both the structural and process models. The Strider tool also allows the user to make alteration and additions to the data originally acquired by Scavenger. Any changes made to the data are written back to the database in the form of XML documents which are consistent with Scavenger format. The Strider tool augments the abilities of Scavenger to provide more specialised graphical editing facilities for the specification of components and processes. Figure 6 below illustrates some of the graphical representation and editing facilities of the Strider support tool.

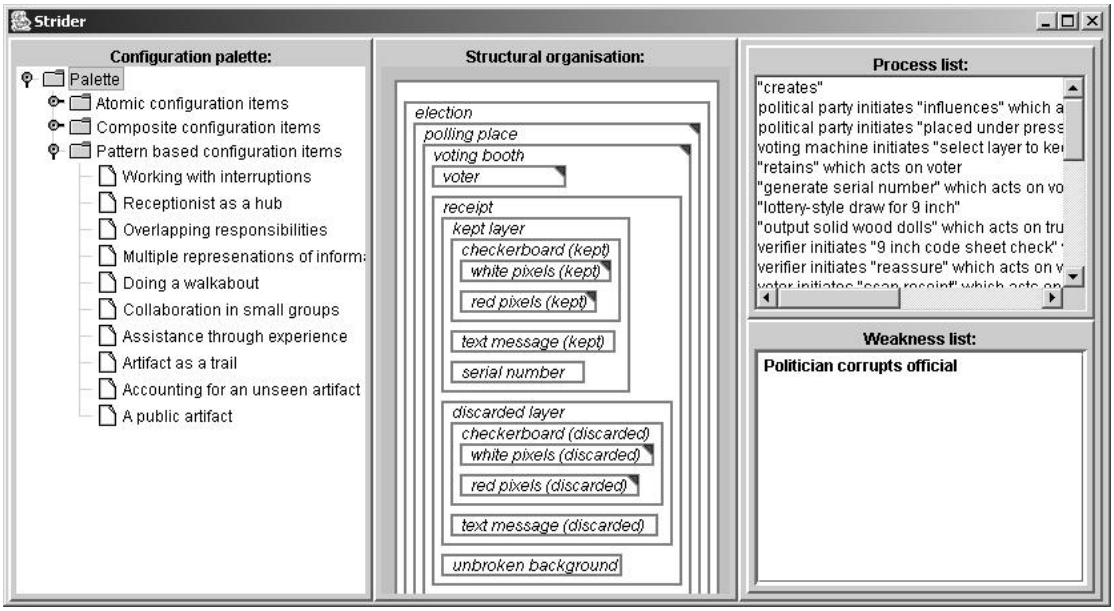

Fig. 6 Strider support tool

Strider provides a number of different types of automated analysis to aid the user in gaining an insight into the dependability attributes of a particular socio-technical system configuration. The automated analysis mechanisms include facilities to allow the user to determine if a specified objective is achievable, the paths which are possible to achieve an objective, the overall patterns

of such paths, the dependencies of these paths and the existence of weaknesses or anomalies within a configuration. Reports for these forms of analysis are generated in HTML and may be browsed and investigated using any modern web browser. Strider also utilises the XML to HTML translation features of Scavenger to give tool users access to documentation about each identified entity stored in the database. This is integrated into the reporting mechanisms provide by Strider, thus allowing users to gain detail descriptive and contextual information about entities presented in the analysis reports.

7.3 Assessment

The previous sections of this paper have demonstrated the applicability of Scavenger to both study approaches, providing support for the unique features and requirements of each. It is possible to extrapolate from these findings in order to envisage the wide range of approaches which it is possible to support using the Scavenger toolset. These could include a variety of applications in ethnographic and system development domains, as well as the collation of information in historical investigation and even potentially crime detection and prevention. Any data modelling technique which derives it's fundamental information from a large pool of unstructured, mixed media material can potentially benefit from the Scavenger approach.

In particular, we have observed the ability of Scavenger to deal with a wide variety of different source materials ranging from electronic texts, through graphics and photos, to digitised audio files. We found the creation of database schemata a particularly fast process, simply requiring the data model of the approach to be specified in XML. This process was however found to be error prone as it was easy for omissions, errors and inconsistencies to be introduced. To help prevent this, a parser was created to provide an automated checking mechanism for the specified

templates. Using this parser, it was trivial to verify the correctness of the schema and make any alterations which were required.

Using Scavenger, the subsequent construction of the structured models was both rapid and accurate. As we anticipated, the use of Scavenger offering clear efficiency gains over the manual processing of the original sources. Finally, we have illustrated the ability of Scavenger to translate the XML models which were produced into browseable HTML, as well as the access offered by the Scavenger database to existing tools and processes.

During the evaluation studies, operators complained that they often were forced into creating concrete entities before they were ready to do so. They found that they were required to make decisions before they had adequate information to make a sound judgement. It thus became clear that there was a potential need for intermediate tags of labels to help the operator progress from the raw material to the final structured model. Using these labels, the operators could tag phenomenon and concepts, collect their thoughts and organise source material before making the final decision and creating a high-level entity. These labels do share some similarities with the concept of "codes" from quantitative analysis, but complement the concrete entities which form the Scavenger database schema.

These additional features were easy to incorporate due to the flexibility of Scavenger, through the addition of an extra set of abstract entities into the data model. This resulted in a hybrid approach in which the abstract tags allowed a more fluid and dynamic process of scavenging which more closely supported the actual work practices of the operator. At the same time, the pre-defined schema provided by the concrete high-level entities still provided a consistent

structure to guide the processes of scavenging, simplify browsing by humans and ease importation by existing tools.

During evaluation it also became clear that additional entity management functionality was required to ease the task of scavenging and database population. Due to the fact that analysis and model construction is a fluid, constantly evolving process, an operator may often change their minds about the classification of a particular entity. This may be due to additional information being brought to light or the progressive development of the set of currently modelled entities. To support these factors, a morphing function was added to the Scavenger tool which permitted an operator to convert an entity from one type to another. With this functionality in place during the assessment exercise, an interesting new scavenging strategy was observed. Operators would first make an initial primary classification for a newly created entity, returning later to refine the entity into a more specific type once more information has been obtained regarding it's exact nature and the nature of other entities to which it was related.

In addition to the morphing function, a merge function was also suggested and subsequently integrated into the Scavenger tool. The merging function allowed two different entities of the same type to be merged together to form a single composite entity. This composite would contain all the description material and relationships of the two separate entities. This functionality was required because incomplete information and uncertainty can lead to the identification of a particular concept of phenomenon multiple times. This can often arise due to the disparate sources of original materials which maybe be used in combination during the Scavenging process. In such a situation identical entities may be referred to using completely

differing terminology. It is essential to provide a mechanism to merge duplicate entities to ensure a complete and consistent Scavenger database.

A further issues which was raised during evaluation, was the desire for linkages between different scavenging projects. Particular observations and phenomenon scavenged from one study or project could well be relevant or illustrative for other projects. For this reason, it was suggested that support be provided for the creation of inter-project links. These could take the form of additional high-level relationships which would have the capability to bridge between model fragments from different projects. Although this is not possible achieve with the current version of the method and support tool, such a feature would be a valuable future addition to the Scavenger approach.

It also became apparent during the evaluation phase that projects which involved a large quantity of source materials would be difficult to scavenge due to the sheer size of the document set. Significant issues of scaleability arouse in organising large numbers of original sources. For example, the simple act of browsing through a set of documents in order to find a particular source became problematic with a large amount of material. In an attempt to address such issues it was possible to introduce a set of different source material directories. Each directory could then be used to store a logical grouping of documents, for example all the materials from a particular source. This made browsing and location of particular documents easier, however further, more sophisticated source material organisation and management features are still required if scaleability is to be dealt with adequately.

A related issue of document management also arose from experiences of using the source traceability features of Scavenger. All original sources are traceable back from the entities which

have been created from them. Thus, from a single entity, it is possible to recall all documents from which it was scavenged. A problem arose if the original source document was particularly large, since it became slow and difficult to isolate the exact point in the document from which media was scavenged. This has implications for the pre-scavenging management of original sources since we must ensure that materials are of a suitable granularity for not only scavenging, but backwards tracing as well.

8 Conclusion

The purpose of this paper was to introduce, describe and assess the Scavenger approach for the reclamation and structuring of data from rich media original sources. We first provided a background section to outline the aims of such an approach, investigated existing techniques and provide a consideration of where they were lacking. Following this, we then proceeded to describe our new approach based around an XML data model. A unique and essential feature of this approach is the automated support which is provided for data reclamation from original source material. Core consideration within the approach is given to the automatic translation of stored data into HTML for browsing and investigation as well as the importation of data in existing tools for further processing, analysis and visualisation.

This paper presents an evaluation of Scavenger based upon two usage studies. In each of these, Scavenger was used to pre-processes data from ethnographic sources in order to build a structured database for later use by the customer approach. These two studies were then used to help assess the applicability and utility of the Scavenger approach and demonstrated the attainment of the originally stated aims and objectives. We reflected upon insights gained from this evaluation phase to propose and discuss possible refinements to the approach.

We found the Scavenger approach to be highly flexible and generally useful to a wide range of different customer applications. This results from the flexible user defined schema, the ability to handle a wide variety of media types as well as dealing with original sources in an unstructured format. The approach supports the work of operators but does so in an unconstrained manner so as not to impinge upon their work processes. Finally, the Scavenger approach provides an open database to allow easy access to scavenged data by existing and future investigation, analysis and visualisation tools.

Future work on the Scavenger approach could include the addition of support for video media fragments, as well as the previously mentioned source material management features. Longer time scale work might include the investigation of inter-project linking and perhaps even the collection of a library of existing Scavenger projects to act as reference material for future scavenging. We are currently actively exploring the use of Scavenger as a pre-processor for integration with a number of third party visualisation and investigation tools. Although developed primarily for use with the outputs of ethnographic study, the Scavenger approach is highly generalisable and we are keen to assess the range of domains for which it is applicable.

Bibliography

[1] Microsoft Corporation, “Microsoft OLE programmer’s reference”, Microsoft Press, 1993.

[2] OpenDoc Design Team, “OpenDoc technical summary”, In Apple’s World Wide Developers Conference Technologies CD, San Jose, CA, April 1994.

[3] A.H.F. Laender, B.A. Ribeiro-Neto, A.S. da Silva, and J.S. Teixeira, “A Brief Survey of Web Data Extraction Tools”, in ACM SIGMOD Record, Volume 31, Number 2, pp84-93, June 2002.

[4] B. Adelberg, "NoDoSE: A Tool for Semi-automatically Extracting Structured and Semistructured Data from Text Documents", In ACM SIGMOD Conference on Management of Data, pages 283-294, Washington, 1998.

[5] L. Richards, “Using NVivo in Qualitative Research”, Sage Publications, London, 1999

[6] Qualis Research, “Ethnograph homepage”, http://www.qualisresearch.com/

[7] QSR Software, “NUDIST 4 Overview”, http://www.qsr.com.au/Software/N4/qsrn4.htm, 1998

[8] Scientific Software Development, “Atlas/ti website”, http://www.atlasti.de/

[9] J. Mackie, S. Lock, "Supporting The Development Of Healthcare Systems Through Situation Modelling", proceedings of the 3rd International Conference on the Management of Healthcare & Medical Technology (HCTM'03), 2003

[10] S. Lock, “The Management of Socio-technical Systems through Configuration Modelling”, to appear in the Journal of Human Systems Management, IOS Press, Amsterdam, 2003