Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2007 Proceedings

Americas Conference on Information Systems (AMCIS)

12-31-2007

Investigating Data Integration Using Sequence Analysis and Process Tracking

Joerg Evermann
Victoria University of Wellington

Follow this and additional works at: http://aisel.aisnet.org/amcis2007

Recommended Citation
Evermann, Joerg, "Investigating Data Integration Using Sequence Analysis and Process Tracking" (2007). *AMCIS 2007 Proceedings.* Paper 479.
http://aisel.aisnet.org/amcis2007/479

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2007 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

INVESTIGATING DATA INTEGRATION USING SEQUENCE ANALYSIS AND PROCESS TRACING

Joerg Evermann
Victoria University of Wellington
jevermann@mcs.vuw.ac.nz

Abstract

The paper proposes process tracing as a research method to investigate IS development tasks. The fruitful application of process tracing is demonstrated using a study of data integration, an increasingly important part of system development. Second, the paper proposes new analysis techniques for analysing the process tracing data. Two analysis techniques that are new to IS process tracing research are demonstrated, pattern discovery and Markov modelling. The paper shows how these can serve to explore process tracing data in order to build theory.

Keywords: Process tracing, sequence analysis, behavioural data, cognitive processes

Introduction

Cognitive research with a focus on the individual has an established history in system development research, with studies focusing primarily on the *outcome* of some IS development task. In contrast, comparatively little is known about the cognitive *processes* that individuals engage in while performing IS development tasks. The aim of this paper is twofold. First, it proposes process tracing as a viable research method to investigate IS development tasks and demonstrates this using a study of data integration, an increasingly important part of system development. Second, the paper proposes new sequence analysis techniques to analyse the process tracing data and demonstrates two such techniques that are new to IS process tracing research, pattern discovery and Markov modelling.

Process tracing is a research method that is used to investigate human cognitive processes. In contrast to traditional experiments or surveys, it focuses on the process, rather than the outcome, of a task (Patrick and James, 2004; Todd and Benbasat, 1987). Process tracing is employed in a variety of IS research areas, such as decision making (Biggs *et al.*, 1985; Broder and Schiffer, 2003), decision support (Cook and Swain, 1993; Todd and Benbasat, 1987), marketing (Kirmani and Baumgartner, 2000), and, especially relevant in this context, knowledge-based systems development (Mao and Benbasat, 1998; 2000), object-oriented system analysis (Wang, 1996), and software verification (Hungerford *et al.*, 2004).

Process tracing research can collect both verbal protocol data as well as behavioural protocol data. While verbal protocols yield rich and easy to interpret data with intrinsic meaning, they have been criticized as being intrusive, distracting, and problematic in their veracity and completeness of coverage (Todd and Benbasat, 1987; Patrick and James, 2004). To address the weaknesses of verbal protocols, information about the subjects' actual behaviour is often collected in the form of behavioural protocol data. Behavioural protocol data is easy to collect automatically (Cook and Swain, 1993) and its collection is less intrusive and distracting than that of verbal protocol data. However, the data is not as rich in meaning as verbal protocol data, merely consisting of numbers and timestamps. Consequently, it is more difficult to interpret.

The most common data analysis technique for behavioural trace data is the scoring of collected data to generate aggregate measures, such as means and variances, for a few key items of interest (e.g. Biggs *et al.*, 1985; Broder and Schiffer, 2003; Cook and Swain, 1993; Todd and Benbasat, 1987). While reducing the data in this way allows easier interpretation, it neglects the rich information about event sequences that is collected with behavioural protocols. This sequence data is useful for exploratory theory building work, but more difficult to analyse and interpret. To date, the analysis and interpretation of sequence data has been limited to plotting and visual identification of patterns (e.g.

Hungerford *et al.*, 2004). However, in large amounts of data, such as that arising from complex system development tasks, this is prone to being highly subjective, difficult, and time consuming.
This paper presents techniques to aid in the analysis and interpretation of behavioural protocol data, focusing on pattern discovery and Hidden Markov Models (HMM). Their application to process tracing data is novel and is one of the contributions of this paper. An example process tracing study is conducted to demonstrate the techniques.
The remainder of the paper proceeds as follows. The next section introduces the study used for illustration purposes. Following this, experimental techniques are described. The data is then analysed using different methods, each examining a different aspect of the data. Data analysis using pattern discovery is followed by data analysis using graph techniques, and data analysis using Hidden Markov Models (HMM). The paper closes with a discussion and outlook to future work.

Example: Data Integration in IS development

This section introduces a study on data integration in IS development projects. Note that this study is used for illustration purposes only, the main emphasis is on the process tracing data analysis methods, rather than the results of this particular study. Consequently, the methodology is described in greater detail than the findings and implications of this study for data integration.
System development increasingly encompasses the integration of legacy systems and especially of their data. In database integration, schema matching is the identification of *similar* elements in two or more databases for the purpose of integration (Batini *et al.*, 1986; Rahm and Bernstein, 2001). Depending on the type of elements to be identified, e.g. entity-types or attributes, different notions of *similarity* can be applied. Even for a single type of element, similarity can take different forms. For example, entity-types may be judged similar because of their relationship-types, their attributes, their names, or a host of other criteria. As a result, the problem of schema matching has given rise to a large number of heuristic algorithms and software tools (Rahm and Bernstein, 2001).
Schema matching methods use information such as data types, optionality, or uniqueness constraints of attributes. The overall schema structure, i.e. the relationships between schema elements such as relationship-types between entity-types, foreign-key dependencies or specialization and generalization, can also be used. Information about schema instances (i.e. database content) can be used in addition to, or instead of, schema-level information. Aggregate information, such as value distributions, term frequencies, averages, etc. is computed for table columns and used to identify similar columns. Machine learning techniques such as neural networks and Bayesian learners are used to establish characteristic features of an attribute.
The evaluation of these heuristics compares their results against similarity judgements made by humans. However, our understanding of the similarity judgement process in humans is limited, as no empirical research in this area exists. As a consequence, little guidance can be provided to the further development of adequate heuristics and tool. Hence, we propose the following research question:

What is the process of judging the similarity of database elements?

The answer to this question is useful in determining how database integrators approach the problem, and how they consequently expect software support tools to behave. Hence, knowledge about human cognitive processes allows data integration researchers to improve their data integration methods and heuristics to conform to their human users' expectations.

Methodology

As there is no existing empirical work, but a multitude of heuristics (Batini *et al.*, 1986; Rahm and Bernstein, 2001), the research question, studying the process of judging the similarity of database elements, was approached using a computerized information display board process tracing study. Eight information display boards were developed based on a review of the information used in existing schema matching approaches (Table 1). Each display board showed information about both databases. The design of the information display boards has a critical effect on the type of findings that can be obtained from a process tracing study (Patrick and James, 2004). However, a full review of the data integration literature is beyond the scope of this paper and the reader is referred to the review articles by Batini *et al.* (1986) and Rahm and Bernstein (2001). While each display board could have been broken down further, the granularity was chosen to offer a balance between the level of detailed data that could be collected, and the usability of the experimental software (Patrick and James, 2004). Four data integration experts confirmed the meaningfulness and usefulness of the information shown on the display boards. These experts did not participate in the process tracing study.

Table 1: Information display boards and their coding for pattern discovery

Information Display Board	Code
Aggregate information in data tables	A
Changes to the data (software modules that update data)	B
Constraints and data types	C
Content of the database table	D
Database Behaviour (frequency of data updates, etc.)	E
Schema (relational schema diagram)	F
Source of the data (software modules that insert data)	G
Usage of the data (software modules that select data)	H

Participants and Study Design

Process tracing focuses on in-depth observation of a small number of subjects. Following the recommendations of Todd and Benbasat (1987) and the example of Hungerford *et al.* (2004), 15 data integration professionals were invited to participate, of which 12 completed the experiment. Participants' experience ranged from 1 to more than 20 years in data integration, and between 1 and more than 20 integration projects.
Software was developed that allows subjects to view display boards with information about two fictitious databases by pushing and holding buttons. Subjects were provided with a description of the information that could be called up with each button. The software recorded button pushes and button releases of the subjects. The top of each screen showed the question "How similar are the elements X and Y in the two databases?" with appropriate element names being substituted for X and Y. Below the question, subjects recorded their similarity judgement on a 9-point Likert-scale. Subjects were presented with five scenarios ("runs") each. Subjects were told that each such run was an independent problem/case. For cross-subject comparability, all subjects were presented with the same scenarios in the same order. Six different, randomised, button orders were created so that pairs of subjects received identical button ordering.
The study yielded sequences of information retrieval for five scenarios ("runs") for 12 subjects, i.e. a total of 60 sequences of button pushes and releases, coded according to Table 1. The sequences varied in length between 5 and 24 viewed display boards.
The following sections describe the analysis of this data using novel techniques of pattern discovery and HMM modelling. As pointed out in the introduction to this example study, the analysis of the process tracing data is useful in determining how database integrators approach the problem, and how they consequently expect software support tools to behave. This in allows data integration researchers to improve their methods' conformance to human expectations.

Pattern Discovery

A pattern is a string of symbols (typically letters of the alphabet). A special symbol denotes indeterminate positions ("wildcard", "placeholder"). A pattern is said to match a given sequence ("is supported by the sequence"), if the sequence contains a subsequence with the same symbols as the pattern and arbitrary symbols in the indeterminate positions. For example, the pattern *A.B*, where "." denotes an indeterminate position, is supported by the sequence *CACB*, beginning at the second position. *ACB* is an instance of pattern *A.B* in this sequence. Patterns may overlap. For example, the pattern *A.A* is found twice in the sequence *ACACA*. A pattern is characterized by its length, its specificity (the number of indeterminate positions) and its support (the number of instances or sequences) in the input sequences.

Analysis Technique

The Teiresias pattern discovery algorithm[1] (Rigoutsos and Floratos, 1998a) was chosen for this study. Other algorithms for pattern discovery are MEME[2] (Bailey and Elkan, 1994) and PRATT[3] (Jonassen *et al.*, 1995). While their principles are domain independent, their implementations are specific to the biology domain, and not suitable for this research.

1 http://cbcsrv.watson.ibm.com/Tspd.html
2 http://bioweb.pasteur.fr/seqanal/motif/meme/meme.html
3 http://bioweb.pasteur.fr/seqanal/interfaces/pratt.html

The Teiresias algorithm identifies patterns with conditions on the minimum specificity *L* for sub-patterns of length *W*, and, optionally, minimum support *K*: Every sub-pattern of length *W* must have at least *L* determinate positions, and occur in *K* of the input sequences. The algorithm works in two phases (Rigoutsos and Floratos, 1998b, 1998c). The first phase scans the input sequences for short elementary patterns of the specified length, specificity and support. The second phase, called convolution, builds larger pattern from concatenations of the elementary patterns. Convolution maintains the specified conditions on specificity, length, and support.
The sequence information from the process tracing software was coded alphabetically (Table 1). As each of the twelve subjects completed five runs, the data set comprised 60 sequences. The algorithm's parameters were set to their default settings, so that out of every sub-pattern with length 5 (*W*=5), at least 3 positions must be determinate (*L*=3). The minimum support was 2 sequences (*K*=2).

Table 2: Most frequent patterns (SxRy = Subject x, Run y)

Pattern	Support	Occurrences
D..CB	12	S2R3, S5R3, S6R1, S6R2, S6R3, S6R5, S9R3, S11R1, S11R3, S11R4, S12R3, S12R5
GCB	11	S6R1, S6R2, S6R3, S6R4, S6R5, S11R3, S11R4, S12R2, S12R3, S12R4, S12R5
D.GC	11	S6R1 (x2), S6R2, S6R3, S6R5, S11R1, S11R3, S11R4, S11R5, S12R3, S12R5
AGC	11	S6R1 (x2), S6R2, S6R3, S6R4, S6R5, S7R1, S12R2, S12R3, S12R4, S12R5
A.CB	10	S2R2, S6R1, S6R2, S6R3, S6R5, S9R3, S12R2, S12R3, S12R4, S12R5
D.G.B	10	S6R1, S6R2, S6R3, S6R5, S9R4, S11R1, S11R3, S11R4, S12R3, S12R5
F.D.G	10	S6R1 (x2), S6R3, S6R4, S6R5, S10R1, S11R1, S11R5, S12R3, S12R5
E.AG	10	S6R1 (x2), S6R2, S6R3, S6R4, S6R5, S8R3, S8R4, S12R3, S12R5

A total of 552 patterns were detected, the majority of which (401 patterns) had a support of 3 or less sequences. The eight patterns with highest support are shown in Table 2. To determine the sensitivity of the discovered patterns to variations in algorithm parameters, the algorithm was re-applied with parameters that allow longer and less determinate patterns (*W*=10, *L*=4). A total of 432 patterns were discovered. Overall, the distribution of the second set of patterns with respect to support and length follows those of the first set. No new patterns were discovered. Hence, the algorithm parameters were not changed further.

Findings

The expectation of universal or dominant patterns in similarity judgement was clearly not fulfilled. The analysis of the eight most common patterns (Table 2) shows that all eight form a cluster of overlapping pattern instances, and are mostly found with subjects 6 and 12, indicating that they may be related to button order (as these two subjects received the same button order). Closer examination of instance overlap reveals that all eight patterns are fragments of the longer and more specific *FEDAGCB* pattern, which is the button order for these two subjects. An interpretation of the longest patterns that were found was not attempted. While they were longer than the most frequent patterns in Table 2, with lengths of up to 16, which may indicate complex processes, their support was low. Because they were only found in two of the 60 analysed sequences, it is unlikely that these are important patterns.

Individual Patterns

As there were no universal or dominant patterns, the pattern discovery algorithm was applied to individual subjects' sequences. As expected from the previous analysis, the individual patterns mostly follow the button order in that subject's experimental condition. However, the following patterns do not:

- Subject 1 repeats the pattern *DCD* (content – constraints – content) three times, signalling a close relationship between the two types of information for this subject.
- Subject 6 exhibits the pattern *CB.H* (constraints – changes – . – usage) in all runs but the first. While *CBH* itself is in button order, the subject deviates from this to include additional information, suggesting the importance of this additional information. In two of the four instances, this information was the data source (G), showing that the information about the software modules that insert, update, and select data is mentally grouped together in a holistic model of the application software.
- Subject 7 exhibits the pattern *EFEF* (behaviour – schema – behaviour – schema) four times, including three times during the last run. This alternating pattern is similar to that found for subject 1. The database behaviour may be

easier to understand when it is viewed in the larger context given by the schema, which shows other data elements connected to the elements in question. The larger context may be important in determining the meaning of behaviour for similarity judgement for this subject.

- Other patterns that show the importance of relating information with each other are *EF..D*, *F..DG*, and *F..EF* (partially overlapping to form *F..EF..DG* with two instances). The fact that this information can be held over time, as evidenced by interruptions of the process, may indicate that, rather than keeping individual facts in short-term memory, the subject has an integrated mental model.
- Subject 8 exhibits a single pattern that is not in button order, comprising *BAGF* (changes – aggregate information – source – schema), which occurs in runs 3, 4, and 5. As this pattern occurs at the beginning of the run, it may indicate that this information is judged important and may have a large impact on the similarity judgement for this subject.
- Subject 9 shows three patterns. The pattern *ADAD* (aggregate information – content – aggregate information – content) indicates that the subject does a consistency check of the given content against the provided aggregate information. The patterns *AEB* and *D..EB* overlap in two instances to form *D.AEB* (content – . – aggregate information – behaviour – changes). Similar to subject 7, where behaviour (E) featured prominently, this data from subject 9 supports its importance.
- Subject 12 shows a short pattern at the beginning of four out of five runs that deviates from button order (*DFE*), strongly suggesting that these are considered the important pieces of information: content – schema – behaviour.

To summarize, this mode of analysis showed that there are no universal or dominant patterns; instead, subjects mostly followed the given button order. Only a few subjects showed individually dominant patterns, while the remainder appear to be easily led into accepting the presented information in any form. Hence, data integration research and applications need to pay attention to potentially influential factors that may appear to be meaningless but clearly influence a subject's judgement process. Moreover, the results suggest that people have no preferred sequence of information retrieval for data integration and there is no set of universally important information that would always be retrieved first.

Pattern Graph

As many of the identified pattern instances overlap, a directed graph (Robinson and Foulds, 1980; Gross and Yellen, 2006) was constructed with patterns as nodes and the percentage of instance overlap as edge weights. For example, 9 of the 12 instances of pattern *D.GC* overlap with those of pattern *D..CB*, leading to an edge weighting of 0.75 between the nodes *D.GC* and *D..CB*. Similarly, 9 of the 12 instances of pattern *D..CB* overlap with instances of pattern *D.GC* leading to an edge weighting of 0.75.

The graph was analysed for strongly connected components (Robinson and Foulds, 1980; Gross and Yellen, 2006), and Freeman-Granovetter groups (Freeman, 1992). Then, the graph was symmetrised (i.e. the weights on two reciprocal edges replaced by the maximum of the two) and then dichotomised, by setting edge weights of less than 1 to 0 and those greater than 1 to 1, effectively retaining edges only between very highly overlapping patterns. The resulting undirected graph was then analysed for cliques (Robinson and Foulds, 1980; Gross and Yellen, 2006).

The results of all three types of analysis of the pattern overlap graph converge on clusters of patterns composed of either *EF*, *AD*, *CD*, or *FDG*. The patterns in the clusters indicate what information about the two databases is conceptually related. They suggest that database behaviour and schema (E and F), aggregate information and content (A and D), constraints and content (C and D), and schema, content, and source (F, D and G) are conceptually related. Changes to, and usage of the data (B and H) do not feature in any pattern cluster, indicating that subjects do not conceptually relate them to other information and use them in isolation. The pattern clusters show that database behaviour (E) is interpreted within the context of the database schema (F). Another cluster of patterns is made up of aggregate information (A) and content (D), which are clearly related.

The clustering supports the idea that subjects construct a coherent mental model of each database, rather than compare the databases on individual information pieces. If the latter were the case, no clusters would emerge. The fact that clusters of patterns are defined by only two or three pieces of information suggests that the mental models built by subjects may in fact be sets of multiple, but connected, mental models, formed around the database content.

Hidden Markov Models

Hidden Markov Models (HMMs) are used to match sequences of observations with a probabilistic state model (Cappe *et al.*, 2005). Figure 1 shows an example HMM with three states and four possible observations. The state transition probabilities depend only on the current state, not on prior history. For example, the probability of transitioning to state 2 when in state 1 is $P(S2|S1)$. The model is called hidden, because the states are not directly observable. For example, there is a $P(O1|S1)$ probability of observing Obs1 when the machine is in state 1, and a $P(O2|S1)$ probability of observing

Obs2. The Baum-Welch algorithm is used to infer the state transition and observation probabilities from a given sequence of observations (Baum *et al.*, 1970).

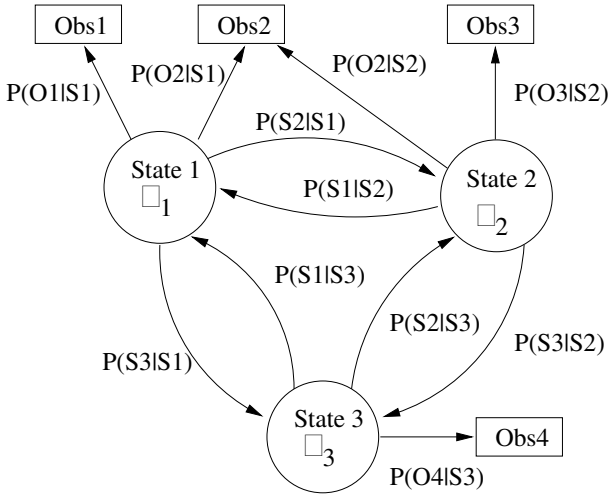


Figure 1: Example Hidden Markov Model

HMMs may be applied to the analysis of process tracing data in two ways. First, when a theoretical foundation or model is lacking, a trivial HMM model can be applied to the sequence data. In such a trivial HMM, each state corresponds to an information display board, and is directly observable. Second, a theory-based model may be used. In this case, the theory will motivate a particular state model and observational probabilities. The observed data is then used to estimate the state-transition and state-observation probabilities.

Trivial HMM

A trivial HMM was applied to the sequence data. The estimated model parameters are shown in Table 3. Examining the transitions from aggregate information (A), there is a probability of 0.459 to transition to viewing source information (G). This probability is larger than would be expected based on button order: Only in 2 of the 6 experimental conditions does source follow aggregate information in button order. Similarly, the probability of viewing content after aggregate information is larger than would be expected from button order. These results confirm those of the pattern discovery analysis, where aggregate information (A) and content (D) form a cluster of related information. From viewing change information (B), the large probability to transition to viewing usage information (H) (0.443) is expected based on button ordering, because in 3 of the 6 conditions change information (B) is followed by usage (H). Similarly, the large transition probability from constraints (C) to changes (B) is expected based on button ordering (3 of 6 conditions).
Focusing on the most likely state transitions ($p > 0.02$), a process model can be developed, shown in Fig. 2. The drawing of the model was begun in alphabetical order, i.e. with aggregate information. However, the actual initial state in this process is each subject's first viewed information display board.
The state transition probability matrix forms a directed graph, and graph algorithms can be applied. This analysis ignores transition probabilities below a certain arbitrary threshold, set here to *0.1*. Table 3 shows that this threshold "thins" out the state transition graph considerably, as 33 of the 64 entries fall below the threshold, and consequently makes the resulting graph easier to analyse. Analysis of the graph yields the following three strongly connected components:

- Constraints – content
- DB Behaviour – schema – usage
- Aggregate information – source

Table 3: Baum-Welch estimated state transition probabilities (all subjects)

From/To	Aggregate Info	Changes	Constraints	Content	DB Behaviour	Schema	Source	Usage
Aggregate Info	0.001	0.000	0.036	0.311	0.095	0.066	0.459	0.033
Changes	0.069	0.061	0.077	0.062	0.176	0.035	0.077	0.443
Constraints	0.019	0.424	0.015	0.124	0.194	0.077	0.120	0.025
Content	0.247	0.006	0.193	0.069	0.018	0.181	0.159	0.127
DB Behaviour	0.249	0.243	0.007	0.209	0.076	0.116	0.085	0.016
Schema	0.205	0.134	0.240	0.051	0.234	0.103	0.004	0.020
Source	0.188	0.025	0.250	0.048	0.061	0.313	0.013	0.102
Usage	0.000	0.142	0.285	0.019	0.160	0.031	0.233	0.131

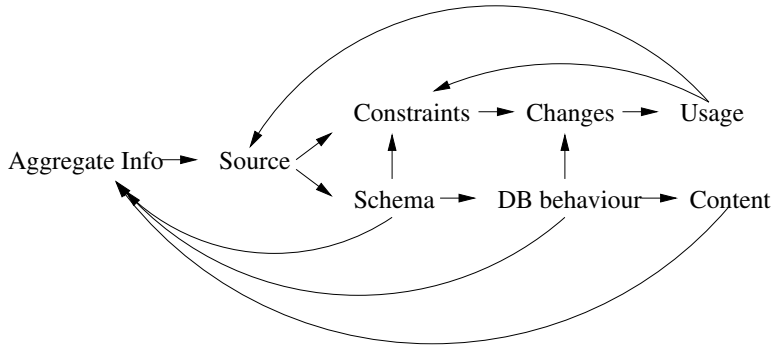


Figure 2: Process model from HMM

The first two components mirror two groups of related information found using pattern discovery in the previous section. There, we identified constraints (C) and content (D) as one group of related information, and behaviour (E) and schema (F) information as another. The component consisting of aggregate information (A) and source (G) is a new grouping found here. The fact that two very different kinds of data analysis methods converge to the same result adds validity to both analysis methods.

Towards Theory

To show the non-trivial use of HMM, a theoretically motivated model is presented. Based on the previous analysis, this HMM summarizes the clusters of related information in the hidden states, whereby these states can yield different observations. For example, the pattern analysis showed that database behaviour (E) and schema (F) often co-occur. Hence, these two observations are related to a single state ("Dynamics") by a probability of 0.5. Five states were defined, and the observations were assigned to the states with equal probabilities:

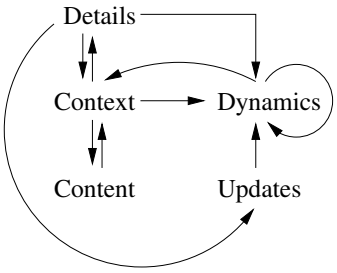
- State 1 ("Dynamics"): DB behaviour, schema, and usage
- State 2 ("Details"): Constraints content
- State 3 ("Context"): Schema, content, source
- State 4 ("Content"): Aggregate info, content
- State 5 ("Updates"): Changes

Table 4: State transition probabilities (reduced HMM, all subjects)

From/To	Dynamics	Details	Context	Content	Updates
Dynamics	0.246	0.163	0.275	0.175	0.159
Details	0.280	0.053	0.273	0.116	0.278
Context	0.371	0.222	0.125	0.230	0.052
Content	0.180	0.140	0.608	0.071	0.000
Updates	0.565	0.095	0.180	0.094	0.066
Initial Prob	0.183	0.160	0.535	0.122	0.000
Limiting Prob	0.309	0.152	0.275	0.156	.0107

The Baum-Welch estimated state transition probabilities, based on the sequence data for all subjects, are shown in Table 4. The row labelled "Initial Prob" is the estimated probability of the state being the initial state for the HMM.
With the transition probabilities estimated, the Markov chain at the centre of the HMM can be analysed for long-term behaviour. As the estimated transition matrix (Table 4) is not doubly stochastic, and is regular, a limiting distribution over the states exists and is shown in Table 4 in the row labelled "Limiting Prob". We can see that in the long term, about 30% of the retrieved information concerns "Dynamics", whereas only 10% concerns "Updates".
In the case that the resulting Markov chain is not regular, states can be classified in three steps (Bhat and Miller, 2002): (1) Identification of irreducible equivalence classes, (2) classification of equivalence classes (or individual states) as being transient or recurrent, (3) determination of the period for each class. Periodic and transient states may be of particular interest to theory development. However, the majority of Markov chains from HMM analysis are regular and therefore irreducible, recurrent and aperiodic.
Examining the most likely state transitions ($p > 0.2$) leads to the state diagram shown in Figure 3. One can see the central role that the state "Context" plays. Subjects are likely to begin in this state (P=0.535), and transitions to other states ("Dynamics", "Details", and "Content") are most likely followed by a return to "Context". However, some other transitions out of state "Details" are also likely, to state "Dynamics" and state "Updates". For further analysis, the sparse graph in Figure 3 can be further analysed for components. In this case, there is only one component.

Figure 3: Likely state transitions, reduced model



This section has confirmed the results of the pattern discovery analysis and identified the same major groups of related information, and identified an additional cluster of related information. Furthermore, by developing a process model of the similarity judgement process, it showed how theory can be built from sequence data analysis. As any other theory, this is a candidate theory that must be subjected to more testing or confirmation with different data sets. Also, the limits of its applicability must be explored by testing it under different conditions, e.g. by varying task or subject characteristics.

Discussion and Conclusion

This paper has shown how process tracing research can provide insight into cognitive processes in system development, in this case the process of data integration. This study was used for illustration purposes only; the main emphasis is on the process tracing data analysis methods, rather than the results of this particular study. Consequently, the methodology is discussed in greater detail than the findings and implications of this study for data integration. The sequence information and the findings from its analysis are not intended to solve data integration problems. Instead, \the knowledge about human cognitive processes in data integration allows data integration researchers to improve their data integration methods.

The paper has shown that advanced sequence analysis techniques can be fruitfully applied to process tracing research. Data analysis in process tracing is carried out in three stages, consisting of scanning, scoring, and theory development (Todd and Benbasat, 1987). Previous research either stopped at scoring data and collapsed the detailed behavioural trace information to simple aggregates, or did not collect behavioural traces at all. The techniques presented here can help the researcher to retain and make use of the information contained in behavioural traces.

The two main methods presented in this paper, pattern discovery and Hidden Markov Modelling (HMM) examine the problem from related perspectives. The pattern discovery focuses on the information viewed by subjects, while the HMM analysis focuses on the transitions between viewed information. Both techniques should be used, to cover both perspectives on the problem.

Both the pattern discovery and the HMM analysis found similar groups of conceptually related information in the data. The fact that two very different kinds of data analysis methods converge to the same result adds validity to both analysis methods.

In contrast to pattern discovery, HMM analysis requires the structural specification of a state machine. In this paper, given the absence of theory, the analysis for the example study necessarily began with a trivial model. However, the requirement of an explicit model is also a strength of HMM. When theory exists, it can easily be accommodated in the form of the structural state model and HMM analysis can be used to estimate free parameters of the theory. In this paper, a provisional theory was derived from the clustering of related information, and the result of the HMM analysis is a process model for judging the similarity of database elements.

The HMM analysis technique can be used for theory building, the final step in exploratory process tracing (Todd and Benbasat, 1987). Without analysis of sequence data, any theory must come from other areas, because aggregate information, such as means or variances, is not rich enough to suggest theory. In contrast, the presented HMM techniques allow theory to emerge from the actual data, as shown with the final HMM example presented here. As any other new theory, theories emerging in this way from process tracing studies are candidate theories that must be subjected to more testing or confirmation with different data sets. The limits of their applicability must be explored by testing them under different conditions, e.g. by varying task or subject characteristics.

Both presented methods can also be used with existing theories. In the case of HMM analysis, theory is expressed in a suitable HMM state model. To accommodate theory in pattern discovery, the observations must be coded according to the theoretical concepts they are hypothesized or known to express.

Some difficulties remain. The behavioural protocol data contains information not only about what, but also about when an event occurs. The coding used here neglects the temporal data and maintains only sequence information. One way to address this problem is to unitise the temporal information, e.g. to a one second interval. For example, a sequence normally coded as *AB* might then be coded as *AAAAAPPBBB*, where 'P' is used to indicate a one second pause. While this type of coding maintains duration information, it can be misleading: In the context of HMM analysis, it suggests that the subject repeatedly transitions from state A to state A, which is not the case. The researcher must weigh the increase in information against the meaningfulness of the resulting analysis. For example, is a two second pause significant? Or, is the difference between two and four seconds of information viewing theoretically significant? The answers to such questions depend on the research question and existing theory.

While HMM modelling can be an important and effective analysis tool, the underlying Markov assumption is not always justified. In the example of this paper, the Markov assumption says that the probability of viewing an information display board (or being in one of the proposed states) depends only on the information display board that was viewed immediately before (or the state the subject was in immediately before). Specifically, the prior history of the subject's behaviour is not relevant. Whether the Markov assumption is justified, is a matter of existing theory and knowledge of the field.

In summary, this paper has presented two novel techniques for process tracing studies, implementations of which are readily available. Both should therefore be among the standard tools of the process tracing researcher.

References

Bailey, T. L. and Elkan, C. "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," In Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, 1994, pp. 28–36.

Batini, C., Lenzerini, M., and Navathe, S. "A comparative analysis of methodologies for database schema integration," ACM Computer Surveys (18:4), 1986.

Baum, L., Petrie, T., Soules, G., and Weiss, N. "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," Annals of Mathematical Statistics (41:1), 1970, pp. 164–171.

Bhat, U. N., and Miller, G.K. (2002). Elements of Applied Stochastic Processes, 3rd edition. Hoboken, NJ: John Wiley & Sons, 2002.

Biggs, S. F., Bedard, J. C., Gaber, B. G., and Linsmeier, T. J. "The effects of task size and similarity on the decision behaviour of bank loan officers," Management Science (31:8), 1985, pp. 970–987.

Broder, A. and Schiffer, S. "Bayesian strategy assessment in multi-attribute decision making," Journal of Behavioural Decision Making (16:3), 2003, pp. 193–213.

Cappe, O., Moulines, E., and Ryden, T. (2005). Inference in hidden Markov models. New York, NY: Springer Verlag, 2005.

Cook, G. J. and Swain, M. R. "A computerized approach to decision process tracing for decision support system design," Decision Sciences (24:5), 1993, pp. 931–952.

Freeman, L. C. "The sociological concept of ”group”: An empirical test of two models," The American Journal of Sociology (98:1), 1992, pp. 152–166.

Gross, J. L. and Yellen, J. Graph Theory and its Applications. 2nd edition. Boca Raton, FL: Chapman and Hall/CRC, 2006.

Hungerford, B. C., Hevner, A. R., and Collins, R. W. "Reviewing software diagrams: A cognitive study," IEEE Transactions on Software Engineering (30:2), 2004, 82–96.

Jonassen, I., Collins, J. F., and Higgins, D. "Finding flexible patterns in unaligned protein sequences," Protein Science (4:8), 1995, 1587–1595.

Kirmani, A. and Baumgartner, H. "Reference points used in quality and value judgements," Marketing Letters, (11:4), 2000, pp. 299–310.

Mao, J.-Y. and Benbasat, I. "Contextualized access to knowledge: Theoretical perspectives and a process tracing study," Information Systems Journal (8), 1998, pp. 217–239.

Mao, J.-Y. and Benbasat, I. "The use of explanations in knowledge-based systems: Cognitive perspectives and a process tracing analysis," Journal of Management Information Systems (17:2), 2000, pp. 153–179.

Patrick, J. and James, N. "Process tracing of complex cognitive work tasks," Journal of Occupational and Organizational Psychology (77), 2004, pp. 259–280.

Rahm, E. and Bernstein, P. A. "A survey of approaches to automatic schema matching," The VLDB Journal - The International Journal on Very Large Databases (10:4), 2001, pp. 334–350.

Rigoutsos, I. and Floratos, A. "Combinatorial pattern discovery in biological sequences: The Teiresias algorithm," Bioinformatics (14:1), 1998(a), pp. 55–67.

Rigoutsos, I. and Floratos, A. "Motif discovery without alignment or enumeration," Proceedings of RECOMB’98, New York, NY, 1998, pp. 221–227.

Rigoutsos, I. and Floratos, A. "On the time complexity of the Teiresias algorithm," Research report RC21161(94582), IBM Research Division, T.J. Watson research center, 1998(c).

Rigoutsos, I., Floratos, A., Parida, L., Goa, u., and Platt, D. "The emergence of pattern discovery techniques in computational biology," Metabolic Engineering (2), 2000, pp. 159–177.

Robinson, D. and Foulds, L. Digraphs: Theory and Techniques. New York, NY: Gordon and Breach Science Publishers, 1980.

Todd, P. and Benbasat, I. "Process tracing methods in decision support systems research: Exploring the black box," MIS Quarterly (11:4), 1987, pp. 493–512.

Wang, S. "Toward formalized object-oriented management information systems analysis," Journal of Management Information Systems (12:4), 1996, pp. 117–141.