In Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centgric Computing, Texas, USA

A Suite of Visual Languages for Statistical Survey Specification

Chul Hwee Kim[1], John Hosking[1] and John Grundy[1, 2]
Department of Computer Science[1] and Department of Electrical and Computer Engineering[2], University of Auckland, Private Bag 92019, Auckland, New Zealand
{ckim001@ec | john-g@cs | john@cs}.auckland.ac.nz

Abstract

We describe SDL, an integrated suite of visual languages aimed at supporting the process of designing statistical surveys. SDL comprises four diagrammatic notations: survey diagrams, survey data diagrams, survey analysis diagrams and survey process diagrams. A proof of concept environment supporting SDL is also presented, together with a cognitive dimensions evaluation of that environment and a cognitive walkthrough evaluation with a target end user – a professional statistician. These demonstrate the utility of SDL and lead us to propose development of a more comprehensive environment supporting the entire statistical survey process.
Keywords: statistical surveys, visual language, visual environment

1. Introduction

Statistical surveys are a common tool for obtaining trustworthy information about a set of objects comprising a target population. The goal of a survey is to describe the population by one or more parameters defined in terms of measurable properties. This in turn requires a *frame,* providing access to the population (eg a phone book or electoral roll), and a method for sampling from that frame [1]. Figure 1 illustrates the typical iterative process involved in defining and executing a statistical survey.

In addition to the survey process, other characteristics of a survey that need modelling include survey data, data analysis and relationships between process, data and analysis. Many tools support aspects of the survey process including SurveyCraft [14] and Blaise [15] for questionnaire design and response collection, SAS [12] for complex data analysis and SPSS [13] for general statistical process design. However, available tools are typically narrow in their focus and there is no agreed notation for defining a statistical survey overall.

Our aim was to design a set of visual notations for statistical survey design that fill a similar role as the Unified Modelling Language (UML) [8] does in software design. Through these notations we aim to:

- make statistical survey design more accessible.
- improve the speed of implementing statistical surveys
- provide better tool support for survey designers

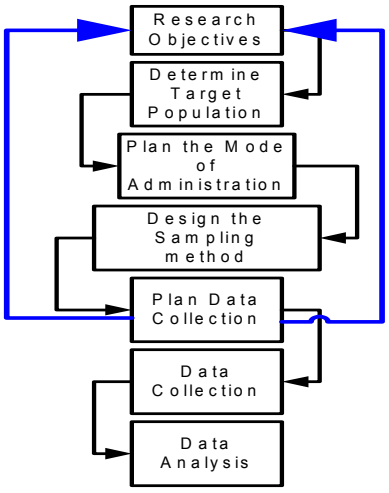

Figure 1. Basic statistical survey process (from [1]).

In the following we describe the background to our work and our design approach before describing and evaluating the Survey Design Language (SDL), summarising our results and describing future work.

2. Background and Requirements

Statistical surveys have become an essential quantitative information gathering and analysis tool in a huge variety of domains. Statistical data is used in the development of products, the analysis of organisational and government performance, the understanding of audiences for TV and radio, political and economic commentary and decision making, and teaching and research within Universities. Designing and implementing a good statistical survey requires a range of skills and experience, and ideally good survey design support tools.

Current survey supporting tools can be generalised into three categories: Computing centric tools, statistical applications and interviewing/data collection aids. Computing centric tools, which are almost synonymous with statistical computing, build on domain specific languages such as S and execution engines. They provide an excellent numerical computing environment, but their steep learning curve and the high level of investment in

low-level implementations required can offset their advantages. Statistical applications provide an environment where users can be insulated from the low-level complexity by introducing a user centred interface for setting up high-level activities. This approach helps users focus their efforts on carrying out the survey process itself, however the proprietary lock-in of operational procedures and back end functionality which overlaps with the computing centric tools, where statistical applications lack the power of dedicated computing tools, are notable tradeoffs. Interviewing/data collection aids focus on the early stages of the survey process thus their benefits do not flow into other parts of the survey process. Thus, while existing survey tools are generally useful they all tend to address only specific parts of the survey process. This forces users to weave a set of tools that lack a semantics to describe the overall survey process.

Our approach to designing SDL was informed by several overlapping methodologies proposed elsewhere. The first, Liu and Liebermann's approach of extracting semantics from natural language [7], was applied by examining a corpus of existing surveys and extracting key design elements and relationships. This helped establish an ontology for survey design expressing key "building blocks" for statistical surveys. We used this ontology to provide the key elements for our SDL visual languages.

The second approach was the close relationship we observed between the requirements of survey design and those leading to process centred software engineering environments [4]. Surveys are very process-centric, with each stage in a statistical process having associated resources (data), processing agents (human and machine), and so on. This suggested a process oriented viewpoint would be an important component of SDL and would be used to link elements from data and analysis viewpoints.

A user centric approach was our third methodological tool [2]. Task analysis of experienced statisticians designing surveys showed survey construction often progresses "bottom-up" from a set of loosely connected goals and analysis tasks. From this we realised SDL must strongly support "brainstorming" for determining survey objectives and survey design refinement. Finally, we were strongly influenced by Burnett et al's guidelines for robust visual language development [3], particularly their four ideal characteristics for visual languages: fewer concepts, explicit depiction of relationships, a concrete programming process and immediate visual feedback.

Combined, these methodologies suggested the following high level requirements for SDL:

- Support visual modelling and management of complex surveys
- Match the user profile, ie people familiar with survey design but who are not necessarily professional statisticians, by making it simple to use and learn.
- Provide a set of integrated notations to allow multiple perspectives to be expressed, including process oriented and objective brainstorming viewpoints
- Present a small number of concepts in each diagrammatic notation
- Explicitly visualize relationships.
- Provide precise semantics for each notation
- Establish multiple levels of abstraction to assist expressiveness.

3. SDL

Based on the methodologies and requirements outlined in the previous section, we developed a design for SDL that supports four diagram types, one of which supports two levels of abstraction. These are:

- Survey diagrams: providing an overview of a survey
- Survey data diagrams: describing at two levels of abstraction the structure of survey data and operations that construct and transform this data
- Survey analysis diagrams: defining statistical procedures and the processes they are composed from
- Survey technique diagrams: defining in more detail a statistical technique's task sequencing /dependencies.

In the following we describe each diagram type in more detail using a common survey design task - a survey of the TV viewing habits of University students. This is an abridged and modified version of a survey carried out by the Odum Institute [9].

3.1 Survey diagrams

Survey diagrams provide an overview of a survey in terms of the various contexts it is organised by and the attributes of those contexts. It supports interactive brainstorming to identify key aspects of a survey such as its requirements implications, analytical methodologies and time scale. Figure 2 shows a survey diagram for the TV-viewing habits survey. The diagram consists of an icon representing the survey (hexagon), contexts by which that survey is organised (ovals connected to the survey) and a hierarchy of attributes for each context (text boxes connected in a tree by arrow connectors). The survey diagram in Fig. 2 shows us that:

- The objective of the survey is to find out TV viewing habits of undergraduate tertiary students.
- The survey's target population is tertiary students.
- It's implications are identification of relationships between viewing habits and academic outcomes.
- To collect data, students are stratified by regions then students are selected randomly from each stratum.
- If a significant relationship is found between TV viewing habits and academic performance the strength of the relationship will be rigorously studied

using discriminant analysis.

A survey diagram's scope covers the entirety of a survey but expressed at a very high-level. There are no rules as to how abstract or explicitly a context is expanded. The relaxed approach has both advantages and disadvantages due to the varying levels of abstraction across contexts. This is discussed later.

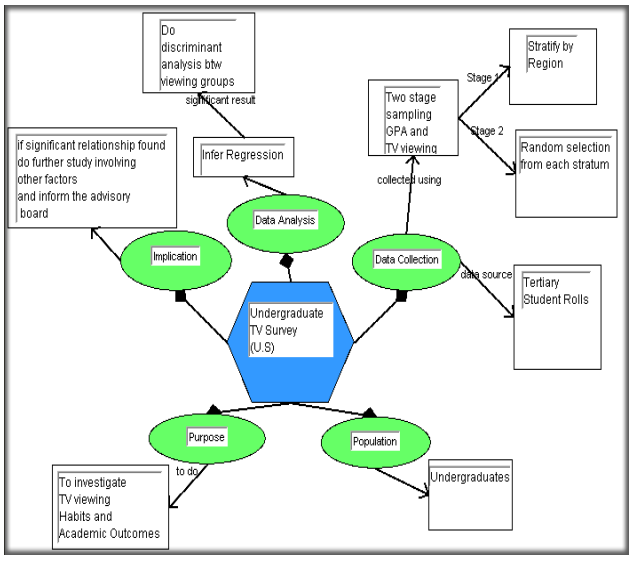

Figure 2. Survey diagram for TV viewing habits.

3.2 Survey data diagrams

Survey data diagrams provide a visual framework to support design of statistical sampling and data collection processes. They represent both the semi structured data structures involved in the survey and sampling operations converting one data structure to another. Two levels of abstraction are supported. Layer 1 is a high level view of data structures and operational flows that connect them. Layer 2 gives more detail of each data structure in the form of a data entity tree, and the particular operations used to sample from the structures. Figure 3 shows notational elements supported in each layer.

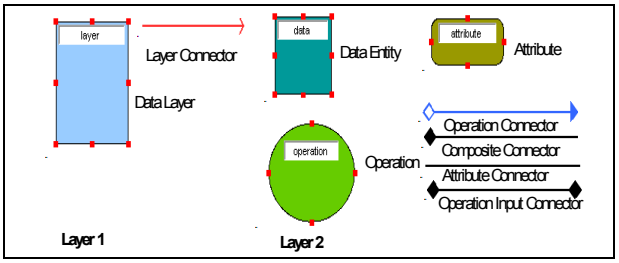

Figure 3. Survey data diagram notation.

Examining Figure 2 we see that for the TV viewing survey, the data source is Tertiary student rolls and that a 2 stage sampling approach is used to select students, firstly by stratifying students by region, and then randomly selecting from each stratum. Figure 4 (1) shows a simplified data entity tree for the data source, while (2) shows a Layer 1 survey data diagram representing the sampling approach. Note how the data entity tree is collapsed into a single icon in this high level abstraction.

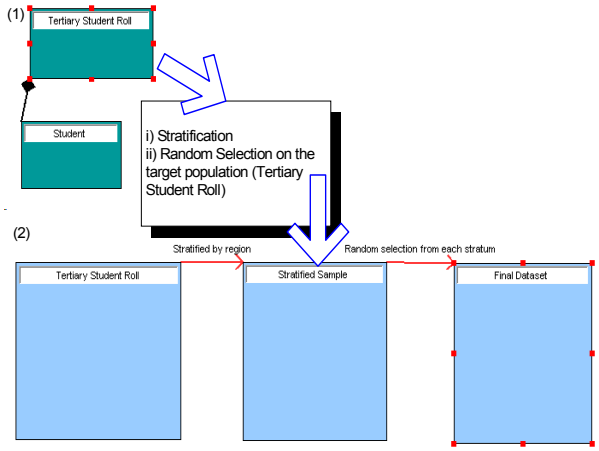

Figure 4. (1) Data entity tree and (2) Layer 1 survey data diagram for TV survey.

This high level description can be refined into a Layer 2 diagram representing finer detail, as show in Figure 5. The dotted boxes are not part of the diagram but show each of the stages represented in the Layer 1 diagram. We can see that a *Stratify* statistical operation is used to group students from the tertiary roll using the *Region* attribute as the stratification parameter (represented by the operation input connector between *Region* and *Stratify*). This produces a new data entity tree in (b). A random selection operation of students within each Region Cell is then used to produce the final data set (c).

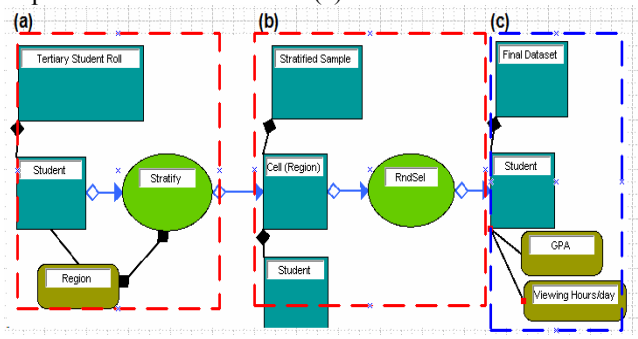

Figure 5. TV Survey layer 2 survey data diagram.

3.3 Survey analysis diagrams

Having defined the high level overview of a survey and the data to be collected, the next step is to describe the analyses to be applied to the collected data using

survey analysis diagrams. These describe in a visual form the statistical processes and techniques used during analysis. The terms process here has a somewhat different meaning than is usual in software systems. A process here is a discrete portion of activity during data analysis to achieve an objective. Examples of such processes include:

- Data cleaning, to remove errors and inconsistency in collected data.
- Handling missing data, dealing with missing responses.
- Data dependency checking, to test the correlation between predictor and outcome variables.

Techniques provide a classical statistical classification of parts of the statistical survey analysis process. A statistical technique is an explicit method employed as part of a procedure. Table 1 shows some examples.

Process	Technique
Missing data handling	Re-weighting Data imputation
Data dependency checking	Regression Multiple regression ANOVA
Data Independence checking	Principal component analysis Cluster analysis

Table 1. Statistical processes and techniques.

Fig. 6 shows a survey analysis diagram for the TV survey. Rectangular icons represent processes and hexagons represent techniques used in those procedures (the association being indicated using a technique connector). Workflow between proceses is represented by process flow connectors (arrows), with the initial analysis step indicated by a circular icon.

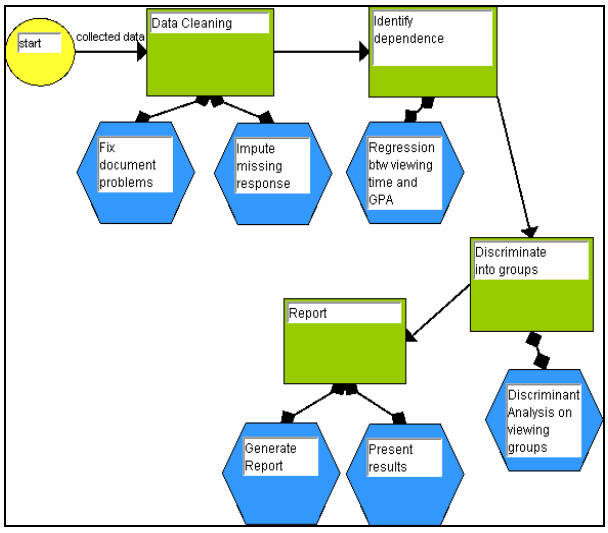

Figure 6. Survey analysis diagram for TV survey.

From Figure 6 we can see that the TV survey analysis has an initial data cleaning phase, which fixes anomalies and imputes missing data. Regression analysis between a student's viewing time and grade point average is used to identify whether there is a dependency between them and, if so, a subsequent process discriminates students into high and low performers to explore this in more detail. The final process produces and displays a report.

3.4 Survey technique diagrams

Survey technique diagrams specify an individual technique in more detail. This diagram, as shown in Figure 7 reuses the data entity tree notation of the survey data diagram and adds information on the technique (logic entity) and the data produced by it (as output ports).

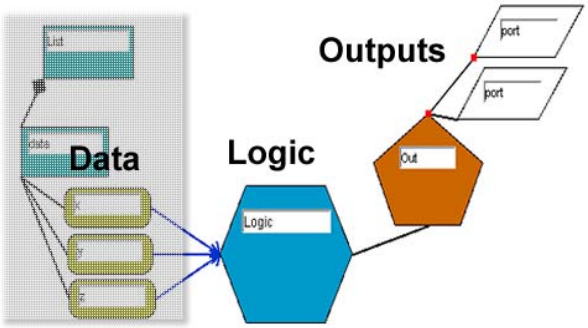

Figure 7. Survey technique diagram structure.

Figure 8 shows its application in the TV survey. Here we see (a) the data entity tree for the sampled data set, (b) part of the survey analysis diagram relating to discriminant testing and (c) a survey technique diagram defining the discriminant text process implemented using a SAS procedure (DISCRIM).

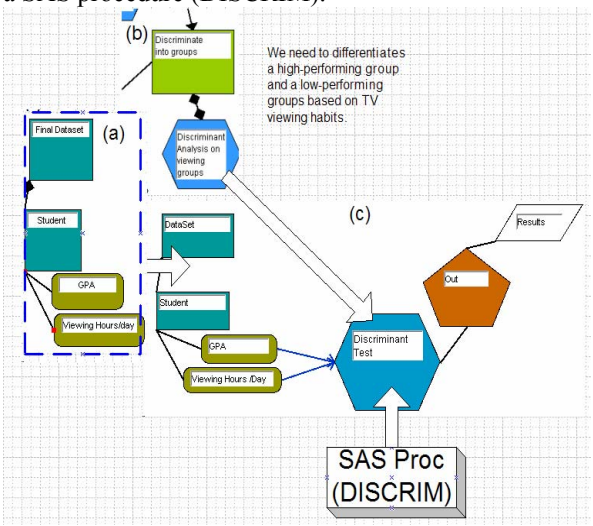

Figure 8. Survey technique diagram for discriminant test.

4. Implementation

We have implemented a set of prototype editors for SDL using the Pounamu meta tool [16]. The SDL diagrams in this paper were generated from screen dumps from these editors. Pounamu allows multi-view, multi-notation diagramming tools to be quickly specified using meta-model, view type, shape definer and event definer meta-tools. Its support for "on the fly" changes to tools allowed us to readily experiment with notational elements and diagram types and modify them at low cost.

We developed a specification of the meta-model for each SDL visual language type capturing the key notational element types. We specified shapes and connectors for representing the elements and then defined a "view type" linking each shape/connector to a meta-model element. Some elements can be shared across SDL view types e.g. survey data specifications. These may be represented in different view (diagram) types with a different (or the same) notational symbol, and map onto the same meta-model element type.

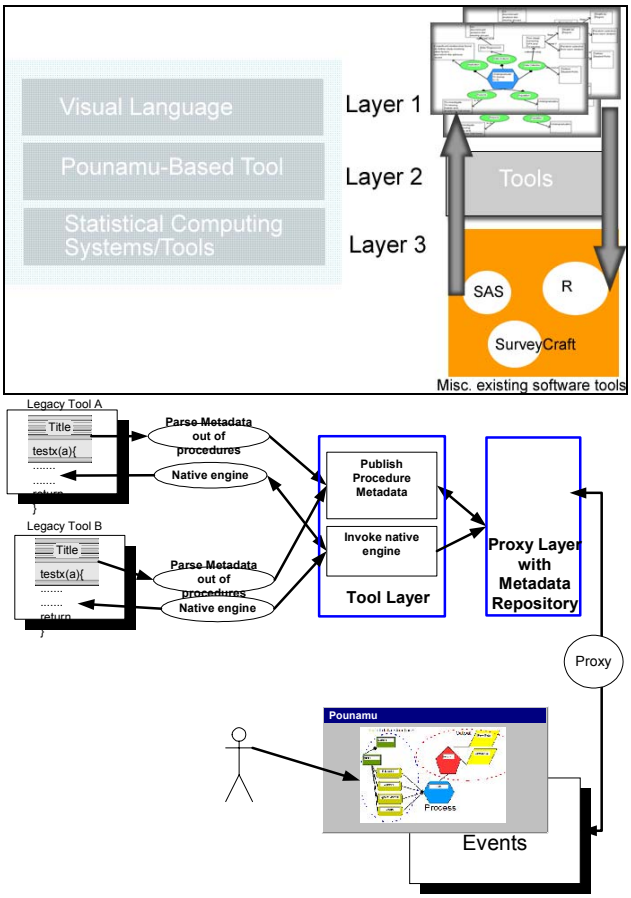

Figure 9. Pounamu SDL tool and statistical tool integration.

Our current work is in integrating our Pounamu SDL prototype with existing survey management tools. Figure 9 illustrates this integration approach. The visual language editors (Layer 1) implemented in Pounamu allow a user to build up a survey design in the Pounamu tool repository (Layer 2). This communicates with existing survey management tools (Layer 3) via both event-based and data exchange-based mechanisms, as appropriate to the external tool. A proxy layer is used to hide the details of the external tool data formats and interaction mechanisms from the SDL editors in Pounamu.

5. Evaluation

The four views of SDL, diagrammatically expressed as four diagrams, appear to fulfil the design guidelines and requirements of section 2. The diagrams can be used to express sample surveys (including the undergraduate TV survey) well, producing a coherent view of the survey. However it is also necessary to evaluate SDL in terms of the usability and suitability of its design. To do this, we conducted two usability evaluations of the Pounamu based SDL tool. The first of these was an extensive cognitive dimensions evaluation [6]. This provided a useful means of evaluating SDL from an end user perspective without involving a large scale usability trial. However, to also address how easy it is to learn to use the tool we conducted a cognitive walkthrough [5] using a doctoral student in statistics as our test subject.

5.1 Cognitive dimensions evaluation

Space constraints prevent us fully reporting our cognitive dimensions evaluation, so we will concentrate on significant features of that evaluation only here.

Survey diagrams are very simple in nature with low abstraction gradient. They afford low premature commitment as contexts and attributes can be added or elaborated on at any time. The concepts represented have good closeness of mapping to the consensus based brainstorming approach statisticians use to map out a survey strategy.

Layer 1 survey data diagrams are also extremely simple with very low viscosity. They do, however, have hidden dependencies which are revealed in the layer 2 diagram. Layer 2 diagrams have higher viscosity. For example data composition processes that modify a data entity tree may have global effect across a diagram. Also, deeply hierarchical data structures can lead to highly viscous diagrams. In practice, however, statistical data sets tend to be fairly flat in structure. The shift of cognitive dimension effects between the two layers is itself interesting, as can be seen from Table 2.

The high level "process" abstractions in the survey analysis diagram allow for low viscosity. This is

important as this diagram is typically used as an opportunistic planning tool. The diagrams as they stand do not allow relationships between processes to be well expressed creating hidden dependencies. We have identified that more work is required to make these dependencies more explicit.

Cognitive Dimension	Layer 1	Layer 2
Viscosity	Low	Can be high if a diagram has many tall data entity trees.
Hidden dependencies	High – but hidden dependencies can be resolved by the use of layer-2	Almost no hidden dependencies unless external operations are used.
Abstraction level	Very high	Data entities are shown at levels of abstraction that are close to underlying low-level features but operations are depicted at a very high level.
Premature commitment	Enforce no look-ahead	Enforce no look-ahead
Abstraction hunger	Abstraction hating	Abstraction-tolerant. Users may introduce new user-defined abstractions. E.g. External operations

Table 2. Cognitive dimensions across layers in survey data diagrams.

Survey technique diagrams show an interesting mix of dimension effects across their three component parts. Technique and output are highly abstracted, but the input data structure is represented at a low level. Thus parts of the diagram relating to the data input structure can be viscous and lead to issues of premature commitment. There is also a hidden dependency between processes and outputs that is not well captured at present. Changing the implementation technique may introduce incompatibility with the output and port specifications. For example a cross-tabulation technique does not work with an output for exporting covariance matrices. More attention to the specification and semantics of typing is needed here.

5.2 Cognitive walkthrough

Our test plan for examining SDL using the cognitive walkthrough approach comprised the following elements:

• Overview of SDL

The test subject was briefly introduced to SDL and working examples explained to observe SDL in action.

• Users Tasks

A list of tasks to be performed by the test subject was given. As the cognitive walkthrough approach focuses on user–oriented solution finding, the tasks were intended to give the test subject opportunities for a self-initiated exploratory path to complete the tasks. Thus the tasks attempted to simulate the cognitive context of a survey researcher in practice rather than imposing fine-grained questions. Three tasks were given to the test subject, all designed to model real-world problems. The first was to request the subject to design a survey diagram for a large-scale UK government sponsored labour force survey [10]. In the second task the subject was given a survey data diagram and requested to explain it and comment on the information represented. The third task involved the subject designing a survey of his choice from scratch.

• User Awareness

A well-designed visual language should give users the sense of self-awareness. In other words, users should be able to tell whether they on the right track in terms of meeting final goals during the course of the using SDL. Insufficient user awareness can especially impact the usability of the diagrams, which are affected greatly by local changes, as late changes could imply a significant overhaul. Therefore the evaluation of SDL looked into not only the final results but also user awareness throughout the testing session.

Task 1 Results

The test subject successfully composed a survey diagram, shown in Figure 10, after a brief introduction and with no interventions. The subject quickly identified survey contexts and added attributes to them. The subject found the notation intuitive and easy to use. Only one small fault was identified in the diagram produced. The attribute in the bottom right corner should have been directly connected to the *Subjects* context rather than to the other attribute as it is not an extension of that attribute.

Task 2 Results

The subject was initially given the survey data diagram of Figure 11. This used an early version of the notation where sampling operations were linked back to the original data structure. From a data type perspective this makes sense, but was found to be confusing by the test subject. This led to the revision shown in Fig. 5, where operations on a data structure results in a new data structure, which should be less confusing foer non programmers. However, we have retained the self-reference type representation as a convenient shorthand

for complex diagrams with the knowledge that this abstraction requires significant learning.

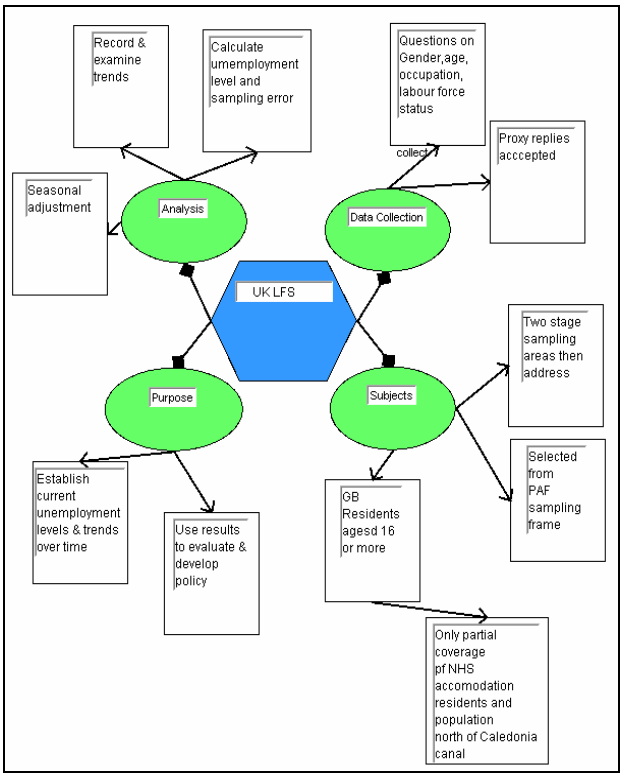

Figure 10. Survey diagram generated by test subject.

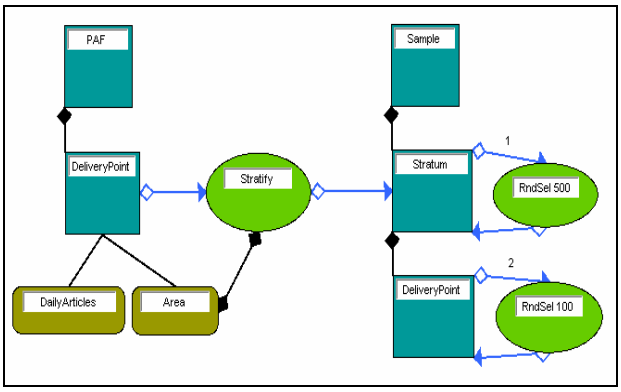

Figure 11. Initial survey data diagram given to subject.

Task 3 Results

Following the first two tasks the subject was asked to derive a survey design from scratch. No specific guidelines were set. The subject chose an analytical survey design based on a harvest estimation survey. The core operational details of the survey were discussed and immediately those details described in SDL diagrams. The survey process included many iterative decision-making components, but these were readily represented in SDL. Figure 12 shows the survey analysis diagram created during the test session. The diagram demonstrates the advantages of SDL in easily turning large amounts of survey design information into a comprehensible visual model that is both clear and has explicit semantics.

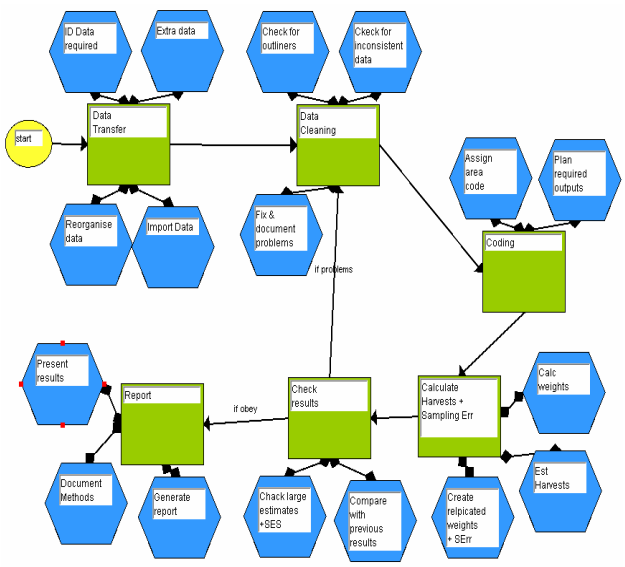

Figure 12. Survey analysis diagram generated by test subject for harvest estimation survey.

The test subject readily and capably applied SDL for modelling the survey in a short space of time and was able to follow rules for association and visual symbols, not repeating the earlier error of misplacing survey attributes. SDL provided a compact description of the survey design describing the activities, including events or items that are required in the execution of the survey process, and associations and revealed the overall purpose and structure of the survey. SDL has a number of limitations identified by the test subject. All of these originated from diagram layout concerns. One example is the survey analysis diagram's tendency to spread out over a large space. It can be more than an aesthetic problem as the spread-out look may slow down a user's comprehension by presenting more visual components than a single visual scan can perceive. One quick solution to resolve the visibility problems is to use multiple views, with each view elaborating on only a limited number of contexts. Another would be making the processes elidable and to build an intelligent layout algorithm to shape the entire diagram to optimise visibility.

6. Conclusions and future work

We have described SDL, a set of visual notations for specifying statistical surveys. SDL aims to provide a similar modelling framework for statistical survey design

as UML does for software design. SDL is comprised of four diagrammatic types, one of which has two levels of abstraction. These together represent both the data and process oriented components of a statistical survey design. A cognitive dimensions analysis and a cognitive walkthrough with a subject expert were undertaken to evaluate the usability of SDL. In addition, we have used SDL to model a large existing survey, Predictor of One-Year Development Status in Low Birth Weight Infants [11], to complement those findings, this practical case study may well be one of most likely circumstances where SDL can be utilised. Surveys rarely become obsolete in to the way software does. Thus revisiting existing surveys is common practice for a survey researcher and SDL diagrams can make rapid review of existing surveys feasible by highlighting core semantics of the surveys. This case study demonstrated all four SDL diagrams in action while also allowing us to investigate a conversion procedure to turn existing survey descriptions into SDL diagrams. Results were highly positive with the survey design was readily able to be modelled and some initial design procedures developed to assist survey researchers to convert existing textual survey designs into SDL.

Several areas of future work have already been identified throughout the paper, specifically better representation of inter-process dependencies, design critics to catch errors such as the attribute attachment error, and multiple view/elision support for large diagrams. In addition, we see considerable scope for providing back end integration with other statistical survey tools so that our SDL environment can be used to not only design a statistical survey, but also implement and control it. Pounamu has an increasingly sophisticated set of integration mechanisms, including RMI and web services based APIs, code generation and import, together with collaborative work support that allows multiple designers to use Pounamu generated tools collaboratively. These can be leveraged to integrate Pounamu with other statistical packages. However, we also see the opportunity to provide a more generic framework for integration using a meta-data based approach for specifying legacy tool capabilities. We are exploring this in current work.

Acknowledgements

The authors gratefully acknowledge the assistance of James Reilly as test subject and statistical consultant, and Nianping Zhu for his work on implementing the Pounamu meta tool.

References

[1] Biemer, P. and Lyberg, L. *Introduction to survey quality* Wiley Inter-Science 2003, Ch 2

[2] Bottoni, P., Costabile M. F., Levialdi, S. , Matera, M., Mussio, P., Principled Design of Visual Languages for Interaction, *Proc IEEE Symposium on Visual Languages '95*, pp. 45-52,1995.

[3] Burnett, M., Baker, M., Bohus, C., Carlson, P., Yang, S., van Zee, P., Scaling Up Visual Programming Languages *IEEE Computer*, March 1995, 45-54

[4] Garg, P. and Jazayeri, M. Process-Centred Software Engineering Environments, *Software Process* Wiley, pp25-49, 1996

[5] Green, T. R. G., Burnett, M. M., A Ko, J., Rothermel, K. J., Cook, C. R., and Schonfeld, J., Using the Cognitive Walkthrough to Improve the Design of a Visual Programming Experiment *Proc IEEE Symposium on Visual Languages*, Seattle, Washington, Sept. 2000, 172-179

[6] Green, T. R. G. and Petre, M. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *JVLC*, 7, 131-174, 1996.

[7] Liu, H. and Lieberman, H. Toward a Programmatic Semantics of Natural Language. *Proc VL/HCC'04* September 26-29, 2004, Rome.

[8] Object Management Group, What is OMG-UML and Why is it important? html://www.omg.org/new/pr97/umlprimer.html

[9] Odum Institute for Research in Social Science, *Computer Administered Panel Study*: University of North Carolina at Chapel Hill

[10] Office for National Statistics (ONS) UK , UK Labour Force Survey Statistical Outputs Group http://www.statistics.gov.uk/STATBASE/Source.asp

[11] Pederson, D., Evans, B., Chance, G., Bento, A. and Fox, A. Predictor of One-Year Development Status in Low Birth Weight Infants *J Dev Behav Pediatr*. 1988 Oct 9(5) p287-92

[12] SAS Institute Inc. http://www.sas.com

[13] SPSS, SPSS Inc. SPSS statistical software http://www.spss.com

[14] SPSS Inc., SurveyCraft, http://www.spss.com/surveycraft/

[15] Statistics Netherlands, *Blaise*, http://www.cbs.nl

[16] Zhu, N., Grundy, J.C. and Hosking, J.G., Pounamu: a meta-tool for multi-view visual language environment construction, Proc *VL/HCC'04* Rome, Italy, 25-29 Sept 2004, pp. 254-256.