

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 1995 Proceedings

Americas Conference on Information Systems
(AMCIS)

8-25-1995

Surveys and Questionnaires: A Markup Language Based Approach

Jeffrey Hsu
Montclair State University

Murray Turoff
New Jersey Institute of Technology

Follow this and additional works at: <http://aisel.aisnet.org/amcis1995>

Recommended Citation

Hsu, Jeffrey and Turoff, Murray, "Surveys and Questionnaires: A Markup Language Based Approach" (1995). *AMCIS 1995 Proceedings*. 148.
<http://aisel.aisnet.org/amcis1995/148>

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 1995 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Surveys and Questionnaires: A Markup Language Based Approach

Jeffrey Hsu, Information and Decision Sciences Department
Montclair State University, Upper Montclair NJ 07043

Murray Turoff, Computer and Information Science,
New Jersey Institute of Technology, Newark NJ 07102

Introduction and Background

An important component of behavioral, social science, and information systems research involves the use of surveys. Questionnaires are an important method for conducting these surveys.

The "traditional" method of creating questionnaires is to plan out the questions on paper, and then type them out on a typewriter or word processor. Creation of survey questionnaires this manual way is cumbersome and slow. The "traditional" method of survey questionnaire design could be referred to as the "direct manipulation" or WYSIWYG (What You See Is What You Get) approach.

The main objective of this effort was to design a set of markups which provides an easier means for creating survey questionnaires, not only paper survey printouts but also automated online surveys. A command language (markup language) approach is used as opposed to the direct manipulation, WYSIWYG approach. While WYSIWYG approaches are generally well suited to novice users, the command language approach often brings about greater efficiency and productivity, especially for more experienced users.

Before proceeding further, it would be useful to define what markups are. Markups are sets of tags, tokens, characters, or specialized commands which are placed into a body of text in order to provide information about the text or other data being processed. (Coombs et al., 1987; Goldfarb, 1981; Wright, 1992).

Markups enable one to "unlock" the data content of a document. In other words, a document is not just a stream of characters, but rather is a data structure which encompasses a great deal of content (Cronk, 1993; Goldfarb, 1991; Van Herwigen et al. 1990).

A descriptive markup (command) language approach allows the creator of a document to define a number of "element types" or "data structures," which identifies a text portion as a member of a certain class. For instance, you can specify if a piece of text is a long quotation, table, paragraph, or a footnote (Coombs et al., 1987; Tuck, 1989; Blake, 1989; MacLeod, 1990; Van Herwijnen et al., 1990).

One important component of any markup language standard should be the inclusion of markups for survey question types. A practical example of an existing markup language standard is SGML (Standard Generalized Markup Language). This standard for marking up text has been defined by various publications (Association of American Publishers, 1992). A widely used subset of SGML is HTML (HyperText Markup Language), which is used to create hypertextbased documents on the Internet World Wide Web.

One of the most important benefits of a descriptivemarkup approach to survey design is its ability to minimize cognitive demands. The use of descriptive markup also has the benefits of better maintainability and portability. In effect, it allows the same data to be used across different applications and platforms, resulting in greater portability. (Coombs et al, 1987; Tuck, 1989; Blake, 1989; MacLeod, 1990)

Survey Generation System Description

The language used in this program is adapted from the Standard General Markup Language (SGML) concept, however it has been modified with respect to syntax to make it easier to input and use. The Markup Language Survey System (MLSS) is a software program designed to enable a user to create, edit, modify, and run online surveys using the markup language system. The program has the following features:

- a) INTEGRATED DEVELOPMENT ENVIRONMENT.
- b) GRAPHICAL USER INTERFACE.
- c) CUSTOMIZED EDITOR/LEARNING FACILITY.
- d) PARSER (SURVEY GENERATOR) MODULE.
- e) HELP SYSTEM.
- f) DATA COLLECTION MODULE.
- g) RUNTIME VERSION.
- h) MOUSE SUPPORT.

Survey Generation Language Description

Question Type: Allocation of Resources

This type of question is designed to allow the user to allocate resources between various choices.

GENERAL FORM

<allocation>

--allocation question text--

<limit> < specifies allocation limit <list /choice1/choice2/choice3/choice4/choice5....>

<allocation> is the question type identifier. This should be at the beginning of the markup and be on a line by itself.

The question text should be placed after the abovetext, not in brackets. The allowed allocation limit comes next. This should be in brackets, such as <2000>.

The <list.....> should follow the previous commands, which consists of an opening bracket, the keyword list, followed by a list of allocation choices, separated by forward slashes (/). The closing bracket completes the command.

EXAMPLE

<allocation>

ALLOCATION OF RESOURCES: You have a total of \$1000.00

<1000>

<list /clothes/vacation/computer/stereo>

will produce the following output:

ALLOCATION OF RESOURCES: You have a total of \$500.00

Press the ESC key when done

clothes ?

vacation ?

computer ?

stereo ?

Total ????

Explicit Response

Frequently, it is desirable to get responses which are of a certain type or format. This kind of item asks for a specific response, such as 'Y' or 'N' or 'T' or 'F'.

GENERAL FORM

<explicit>

(question text)

<explicit> is the keyword for the explicit response question. Question text is placed after the keyword in brackets.

EXAMPLE

<explicit>

Do you drive a car?

Free Response

Often, the need arises to allow the respondent to provide more information than just a single response and allows the respondent to enter lines or paragraphs of text.

GENERAL FORMAT

<free response>

(text of the question)

When this markup is executed, it will display the question, together with an editing screen which allows the user to create a freeform text response to the question.

EXAMPLE

<free response>

What are your thoughts about the state of the world?

Likert Scale

Frequently, the need arises to express various levels of agreement or disagreement with a statement or idea.

GENERAL FORMAT

<likert> --required function name

<[nominal/interval]> --type specifier

--text of the question-- --question text

<endpoints /endpoint1/endpoint2>

<list /choice1/choice2/choice3/choice4>

There are four basic elements to the Likert question markup:

<likert> is the required function name, which specifies what type of question it is. nominal/interval. The nominal type will allow selected choices to be chosen, while interval will allow intermediate values to be selected.

<endpoints / . / . / .> will specify the particular endpoints which will be placed on each end of the scale.

<list / . / . / .> allows you to specify the individual choices which you are allowed to make in regards to the Likert question.

EXAMPLE

<likert>

<interval>

Summer is my favorite time of year.

SA=Strongly Agree, A=Agree, N=Neutral, D=Disagree, SD=Strongly Disagree

<endpoints /strongly agree /strongly disagree>

<list /SA/A/N/D/SD>

which will produce output appearing as follows:

Summer is my favorite time of year.

SA=Strongly Agree, A=Agree, N=Neutral, D=Disagree, SD=Strongly Disagree

: SA : A : N : D : SD :

Strongly Agree Strongly Disagree

Multiple Choice

One of the most commonly used types of survey questions asks the respondent to choose between various options.

GENERAL FORM

```
<multiple choice> --required function name  
--text of the question--  
<(alpha/numeric)(single/multiple) (horizontal/vertical)>  
<list /choice1/choice2/choice3.....>
```

The following are the options for this markup:

<multiple choice> is the required keyword identifier for this question type. The text of the question, without brackets, follows the keyword identifier. The third set of commands specifies the multiple choice options, which includes:

- a) alpha or numeric choices.
- b) single or multiple responses.
- c) horizontal or vertical format.

<list // /...> specifies the choices for each of theselections in the list. The individual choices should be separated by forward slashes.

EXAMPLE/ALPHABETIC. An example of how this markup can be used is as follows:

```
<multiple choice>  
What is your favorite season?<  
alpha single vertical>  
<list /spring/summer/fall/winter>
```

which will produce output as follows:

What is your favorite color?

- a. spring
- b. summer
- c. fall
- d. winter

Choice: []

Rank Order Questions

The markup presented here allows respondents to rank order the choices given on the screen.

GENERAL FORM

The general form of the markup is as follows:

```
<rank order>
--text of the question--
<list /choice1/choice2/choice3/choice4>
```

<rank order> is the required keyword for rank order question markups.
the text of the question follows the questions, and is not placed in brackets.
<list /././....> specifies the choices which you want to specify for the rank ordering.

EXAMPLE

```
<rank order>
Which food would you like best?
<list /pizza/ice cream/hot dog/apple pie>
```

produces output as follows:

```
Which food would you like best?
pizza { }
ice cream { }
hot dog { }
apple pie { }
```

Semantic Differential Scale

GENERAL FORMAT

```
<semantic>
--question text--
<endpoints /endpoint1 /endpoint2>
<list /choice1/choice2/choice3/choice4/choice5...>
```

<semantic> is the required keyword for the semantic differential question text.
the question text follows the keyword <semantic> on a separate line, and is not placed in brackets.
<endpoints /././...> specifies the two endpoint descriptions which are displayed on the screen for that question.
<list /./././....> specifies the individual choices which the respondent will select from when answering the survey.

EXAMPLE

```
<semantic>
Taking an exam is stressful.
1=Agree, 4=Neutral, 7=Disagree
<endpoints /agree /disagree>
<list /1/2/3/4/5/6/7>
```

which will produce the following output:

Taking an exam is stressful.

1=Agree 4=Neutral 7=Disagree

1 2 3 4 5 6 7

Agree Disagree

References

References available upon request.