First Experiences with TMM

Kevin Mayo

TR 93-08

Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

February 24, 1993

FIRST EXPERIENCES WITH TMM

Kevin A. Mayo
Department of Computer Science
Virginia Tech (VPI&SU)
Blacksburg, VA 24060
mayoka@cs.vt.edu

ABSTRACT

The Task Mapping Model (TMM) is a human-computer interaction technique that supports situational analyses to derive new design requirements from formative evaluation findings. While the TMM methodologies and formalisms are currently being developed and validated, this paper reports some informal first experiences and findings. Two developers who are presently working on independent non-trivial interfaces were asked to use TMM and comment on it. The developers found benefit in TMM analyses, albeit in different ways, and their views were captured with an informal subjective post-analysis survey and reported here.

INTRODUCTION

The Task Mapping Model (TMM) is a human-computer interaction technique that supports the interface development process. (Mayo, 1993; Mayo & Hartson, 1993) TMM provides a framework by which new design requirements can be synthesized from formative evaluation findings and task analysis. This model answers the pragmatic concerns of industry (cost effectiveness and impact of modeling) and academia (formal modeling and completeness), and attempts to reassert the usefulness of pragmatic interaction modeling (Hartson & Mayo, 1993)

Briefly, the TMM is a process by which formative evaluation findings isolate usability problems, and TMM analyzes these problems to derive new design requirements. This process is broken down into three phases: (1) isolate, define, and describe a user-task, (2) refine and categorize user knowledge needs from task descriptions, and (3) synthesize new design requirements. (See figure 1.) TMM's overall approach to task modeling is a practical one, and therefore, each of these steps is straightforward and as unencumbered (as possible) by formalisms.

TMM is currently being validated through experimentation involving design and redesign using TMM and another analysis method. The design of this experiment, its analysis, and its results are not our focus here—these will be subsequently reported.

This paper reports on the first experiences using TMM in a design setting. Two participants were chosen—each working on a non-trivial interface (or set of interfaces)—and were asked to use TMM for analysis. Beforehand, the designers were provided with a brief introduction to TMM (Mayo, et al., 1993) and the TMM Analyst Guide (Mayo,

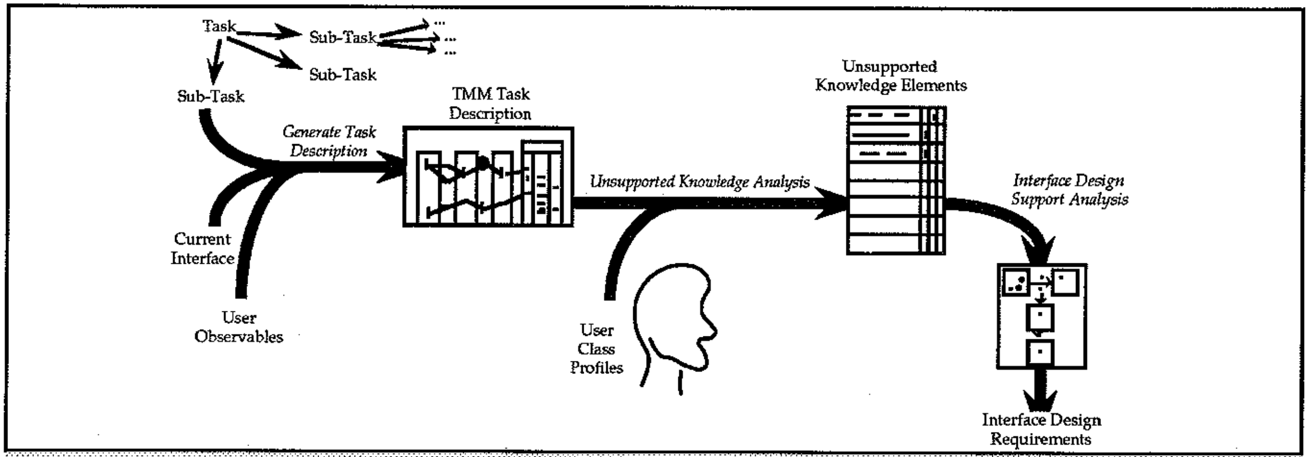
Task
Sub-Task
...
Sub-Task
Sub-Task
Generate Task Description
TMM Task Description
Unsupported Knowledge Analysis
Unsupported Knowledge Elements
Interface Design Support Analysis
Current Interface
User Observables
User Class Profiles
Interface Design Requirements

FIGURE 1: TMM ANALYSIS LIFE-CYCLE

1993). During the analysis, the designers were paired with a TMM expert (the author). In the analyses, the designers served as interface- and problem-domain experts, while the author served as a TMM expert and user advocate. The analysis sessions took three to four hours for each designer, and one designer participated in two sessions. Afterwards, the designers answered a post-analysis survey based on their experiences and observations. The survey is broken down into several collections of questions, and the results are reported here. Also, commentary from the author (who participated as the TMM expert) are presented here as objectively as possible.

PARTICIPANTS

Two interface designers/developers were chosen to participate in this informal initial discovery study. The first designer is a graduate level computer science student with extensive theater experience. This experience includes set design, costume design, and make-up design. This designer employs all of these design experiences while developing interfaces, and the experiences influence: user attention focusing, mixing graphics and text, and graphic design and layout. Even though this designer has taught undergraduate-college level computer science courses, this designer does not have a B.S. in computer science, but looks forward to a M.S. as a first degree—followed by a Ph.D. This designer currently researches in the area of human-computer interaction, but has no industrial experience. This first designer, who we will refer to as **Designer A**, is currently working on the user interface for an interactive database of computer science literature (including, but not limited to, text, audio, visual, dynamic algorithms, and hypermedia.) During the analysis, a small part of this system design was examined using TMM. A second analysis session was done with this designer over a section of the system that has not yet been designed.

The second designer is also a graduate level computer science student. This designer, however, received a B.S. and a M.S. in computer science and subsequently worked for a major computer company. This designer later decided to return for a Ph.D. in computer science, and currently researches in geographic information systems with interests in educational software. While this designer has industrial experience, the designer's knowledge of human computer interaction concepts and techniques is only intuitive—i.e., this designer has had no formal introduction to the field. This second designer, who we will refer to as **Designer B**, is currently working on a series of tools and their interfaces to support interactive learning of various geographical concepts (including, but not limited to, population expansion and migration by area, computer cartography, and region identification.) A single user interface design from these learning tools was selected and analyzed with TMM .

The author served as the TMM expert in the analyses, and will be referred to as the **Supporting Designer**. The supporting designer limited his involvement so as to not induce biased responses from the designers. Typically, the supporting designer translated the designers' responses into TMM formalism, helped probe the task structure from the users' point of view, and guided the TMM analysis process with respect to the TMM analysis life-cycle. The designers had an opportunity to critique the supporting designer in the post-analysis survey.

POST-ANALYSIS SURVEY QUESTIONS AND ANSWERS

The questions in the survey are broken into several categories: TMM task descriptions, unsupported-knowledge analysis, TMM synthesis of design requirements, TMM analysis life-cycle, effects of two people, and other comments. Each category has several questions. Figure 2 shows the questionnaire.

Category: TMM Task Descriptions

The generation of task descriptions within the TMM framework represents the first phase of the TMM life-cycle. The TMM task description framework consists of several abstraction domains (*problem, computer semantic, computer syntactic*, and *articulation domain*), categorized knowledge elements, and mappings describing task

TMM Task Descriptions

- What was most difficult about task description using TMM?
- What was easiest about task description using TMM?
- TMM Task Domains
  - Are the domain distinctions clear?
  - Are all levels of abstraction present? If not, more or less?
  - What problems did you have casting tasks into the domains?
- TMM Task translations and mappings
  - Are mappings obvious and clear? If not, why not?
  - What problems did you have creating the translations?
  - What problems did you have creating the mappings?
  - What problems did you have associating user needs with mappings?
- TMM User Knowledge
  - Are there enough knowledge categories? If not, what else?
  - What problems did you have identifying the user knowledge?
  - What problems did you have categorizing the user knowledge?
- Other comments on TMM Task Descriptions

TMM Unsupported-Knowledge Analysis

- What problems did you have determining knowledge support/non-support?
- Are all categories of support/non-support represented? If not, what else?
- What characteristics should be in the user class profiles?
- How did you compare needs with user class profiles and current interface?
- What problems did you have in the comparison?
- Other comments on TMM Unsupported-Knowledge Analysis

TMM Synthesis of Design Requirements

- Does TMM provide enough information to synthesize design requirements?
  If not, what else is needed?
  If too much, what information is not needed?
- Should TMM provide more than design requirement specification?
  If so, what?
- Other comments on TMM Synthesis of Design Requirements

TMM Analysis Life-Cycle

- Are any stages left out of the TMM analysis life-cycle? If so, which?
- Are the stages distinct enough?
- What do you see as the analysts needs (aside from the stated) for each stage?
- Comments on traditional Task Analysis and TMM
- How does TMM compare with other methods you are familiar with in terms of time, effort, complexity, etc.?
  - Usefulness of TMM as a Task analysis method?
- Time efficiency
  - Is the analysis life-cycle time efficient?
  - How can time efficiency be improved?
- Other comments on TMM analysis life-cycle

Effects of Two-People

- Is TMM is a single or group analyst process?
- What effects did the TMM-expert's presence have on the analysis?
- What effects did Designer's presence have on the analysis?
- Other comments on using TMM in groups/singularly

Other Comments in General

- What other comments do you wish to share concerning TMM?
  TMM's process? TMM's function within Interface Design?
  Effectiveness of TMM? TMM's dependency on user class profiles?

FIGURE 2: DESIGNER POST-ANALYSIS QUESTIONNAIRE

translations within the domains. The first set of questions focused on the overall form, content, and usefulness of TMM task descriptions for analysis.

A common problem both designers had is based on the perceived analysts' role with regards to task descriptions. Designer A felt that more attention to the analysts' role was needed, **A**: "...[I had difficulty in] keeping my roles straight—that is, remembering when to focus on a task instead of the existing design." In other words, should the analyst emphasize task analysis or center on design analysis and interaction specification? This was a problem with Designer B also, but the problem stemmed from the existence of a previous artifact (design prototype), **B**: "I was also biased by the current interface, so envisioning new ones was more difficult." The analysts' roles seem to be confounded with the roles of the designer; yet, an underlying assumption of TMM is that analysts and interface designers are two different groups, or in the very least, a group that is fully cognizant of the dual roles they serve. Both designer **A** and **B**'s confusion in this matter only serves to strengthen the need for separation of analysis and interface design duties.

When asked about the TMM task description framework, both designers proposed extensions to the problem domain. Designer **B** suggested: "...break the problem domain up into 'real world problem' and 'model of problem'", while designer **A** offered: "It might be helpful to deal with the task in ... [a] semantic/syntactic way." The suggestion to divide up the problem domain is not new—initially TMM had semantic and syntactic domains that represented the problem domain. However, while working with TMM task descriptions it became clear that a task semantic and task syntactic domains could be combined into a problem domain; yet, the computer semantic and computer syntactic domains could not be combined because of differences among interface interaction styles.

Also with regards to the TMM task description framework, both designers agreed that articulation domain elements are the easiest to model. This is not a surprising finding because designs that already existed were described using the TMM framework, and the articulation domain (using UAN (Hartson, Siochi, & Hix, 1990)) is based on the previously designed interactions between user and machine. Designer **A** puts it bluntly: "[What was the easiest?] Initially, the articulatory details—because they already existed." But when describing a task that has not been designed yet, designer **A** points out that "...identifying knowledge needed at the upper levels was really pretty easy—surprisingly so, in fact." This is a very welcome observation, as much of TMM's analysis quality relies on the thoroughness of task descriptions which include knowledge needs. The easier these knowledge needs are to identify, the more thorough will be the description and analysis.

As mentioned above, the realization and identification of necessary user knowledge needs within the mappings is paramount to TMM analysis quality. TMM cannot synthesize proper design requirements without correct user knowledge needs. Designer **B** found the categorization of knowledge into *factual*, *conceptual*, and *procedural* elements to be easy and satisfactorily-expressive; yet, designer **B** did voice concerns about the granularity and quantity of knowledge that may need enumeration. This is not an uncommon phenomenon to modeling—how far to go with the description of necessary knowledge? TMM notation allows grouping of factual knowledge elements into a single conceptual knowledge element as a short-hand notation.

Category: Unsupported-knowledge Analysis

The next phase of analysis within the TMM life-cycle is unsupported-knowledge analysis/synthesis. In this phase, all user knowledge needs are enumerated and classified as supported or unsupported elements. A knowledge need is supported if the interface provides access to the knowledge or the user possesses the knowledge (i.e., it is specified within the user class profiles). The supporting designer, the TMM specialist, selected a subset of knowledge needs from each interface task description under scrutiny. This selection was based on the category of the knowledge needs and their support. The supporting designer strove to select a cross section of categories with and without support. The second survey questions set is based on this second phase of TMM analysis.

Designer **B** found it easy to determine knowledge element support because a prototype of the interface design analyzed already existed. On the other hand, an interface design that has no prototype automatically sets all knowledge needs not supported by the user class profile(s) as unsupported. Therefore, it should be straightforward to determine knowledge element support. Designer **A** observes that the determination of knowledge element support can also be backed up by user testing during formative or summative evaluation.

Designer **A** focuses on a different aspect of the unsupported-knowledge analysis. Designer **A** again brings up the question of roles: What is the role of the analyst in this phase? To examine the task structure or the interface design?

An interface designer faced with identifying unsupported knowledge needs might just as easily say, "It's obvious to the most casual observer." Of course, the interface designers can be immersed so deeply within the design that they cannot see unsupported knowledge elements. This is not necessarily the fault of designers, who are only doing their jobs, but it again shows the need for separation of analyst and interface designer duties when ever possible.

Category: TMM Design Requirements Synthesis

The third, and final, phase of analysis in the TMM life-cycle is the synthesis of design requirements from the unsupported-knowledge needs identified in the second phase. The selection of knowledge elements by the supporting designer afforded several instances of synthesizing design requirement for both participating interface designers. The results of the next set of survey questions is based on the designers experiences and observations in this phase.

When asked if TMM analysis provided enough information for new design requirement synthesis, designer **B** wrote: "Yes. The design flaws came out very easily. In fact, the way TMM presents the interface flaws begs for the correct (or at least a workable) solution to the problem."

Designer **A**, when asked the same question, remarked: "Don't force the analyst to deal with the articulatory level until after a design has been completed. As a designer, I'd hate to have that information form the analyst." In TMM analysis only design requirements are synthesized.

The designers were then asked if TMM should synthesize more than design requirements. Designer **B** answered: "No. I view TMM's purpose as exactly that." By deriving only design requirements and not new design specifications, TMM does not tie the hands of interface designers, or stifle creative design solutions.

Category: TMM Analysis Life-Cycle

The three phases of TMM analysis taken together represent the TMM analysis life-cycle (see figure 1). Each phase serves a specific analysis need: first task description, next unsupported-knowledge needs, and finally synthesized new design requirements. This life-cycle is presented in (Mayo, et al., 1993) and both designers were introduced to it. This section reports on questions based on TMM's analysis life-cycle and how it fits into the user interface development process.

Designer **A**, very concerned with analysts producing interface designs, commented on TMM analysis life-cycle and its place within user interface development: "It's fine for analyzing a preliminary design and would help to identify problems in how the user will perceive the syntactic and articulatory domains, but that should not be handed down as a given. Perhaps the designer should work with TMM and pass this level of information back to the analyst." Here designer **A** proposes that analysts not do task descriptions at lower abstraction levels, and again raises the need for involving multiple people/roles during TMM analysis so as to separate duties.

The second designer, **B**, found TMM useful as a task analysis method, but stated "I think it would be most useful during initial design, not afterwards." When queried about time efficiency, designer **B** added: "There is the danger of going into too much detail in task division. Particularly dividing the wrong tasks (non problem areas)."

Category: Effects of Two People

The next section of the post-analysis survey attempted to assess the impact of analysis groups, and the influence of the designer and supporting designer.

Both designers, **A** and **B**, agreed that a single person performing a TMM analysis could easily get "mentally stuck in one domain." They also agreed that the TMM analysis participants should have different perspectives, and designer **B** proposed that "Using TMM requires several viewpoints—the interface designer should be there, a TMM expert, a typical user, and some type of mediator to keep things flowing along." Designer **B** also went on to propose that "interface designer should NOT be at the TMM analysis" for purity of analysis reasons.

The survey also questioned the impact of the supporting designer. Designer **A** felt that the supporting designer forced issues in the upper level abstracted domains on the previously defined interface design, while quickly moving down to the lower abstraction levels on the undefined task. (In fact, the roles were reversed because the supporting designer had an interface design in mind for the undefined user task.) In both cases, the TMM analysis showed problems that were easily overlooked by the designer and supporting designer. This shows how different perspectives can add to the analysis—forcing issues that one person may overlook or trivialize. Designer **B** reported the role of the supporting designer as a TMM expert and also a typical user.

The impact of the interface designer (designers **A** and **B**) on the TMM analysis was also responded to. Both designers approached the analysis as 'interface designers' as opposed to 'analysts'. This is not surprising—performing dual roles and trying to be objective on one's own work is very difficult; often people are too critical of their work or not critical enough. In fact, designer **A** stated "For the first session, [I had] a tendency to stay with the existing design rather than with the task." Designer **B** also had the same reaction: "Unfortunately, the interface was already designed, so I was somewhat reluctant to change it. I was also biased by the current interface, so envisioning new ones was more difficult. Furthermore, any new interface would be implemented by me, so I was hesitant to point out flaws." These are not uncommon reactions by interface designers, and often they fall on sympathetic managers' ears who are concerned with deadlines. In other words, pragmatics can, and often does, win over "ivory-tower perfection".

FINAL OBSERVATIONS

The designers who participated in this informal study had several common important points about TMM. In summary they are:

- TMM is a useful analysis technique to synthesize new design requirements.
- TMM analysis is best served if performed by several people with various points of view.
- TMM analysis results, new design requirements, are only suggestions of unsupported-user knowledge needs that could represent usability problems. They do not, nor should they, indicate any particular design solution.
- TMM analysis can be performed for both initial interface designs (preferred by designer **B**) and for situational analysis of interface designs (preferred by designer **A**). However, the type and style of analysis and results depend of which method is chosen.
- Existing interface designs can hamper the thoroughness of TMM analysis. As designer **A** states, "Once that jump to seeing a design is made, it's damnable hard to get back to the task level and the openness needed for creativity."
- Task descriptions should focus on the higher abstract levels before proceeding to lower levels.

This study is only a starting point for TMM validation. This study shows, in a loose sense, that TMM can be useful in interface development. However, this does not show specific TMM faults or benefits, and a sample size of two designers is far from ideal. This study served to show areas of TMM methodology that need further research and refinement.

ACKNOWLEDGMENTS

This small informal study would not have been possible without designer **A** and designer **B** (who shall remain nameless). I cannot thank them enough for their time, effort, and support.

BIBLIOGRAPHY

Hartson, H. R. & Mayo, K. A. (1993). *Synthesis-Oriented Situational Analysis as an Alternative to Analytic Evaluation for Iterative Interface Design* (TR-93-09). Department of Computer Science, Virginia Tech (VPI&SU), Blacksburg, Virginia.

Hartson, H. R., Siochi, A. C., & Hix, D. (1990). The UAN: A User-Oriented Representation for Direct Manipulation Interface Designs. *ACM Transactions on Information Systems*, *8*(3), 181-203.

Mayo, K. A. (1993). *Task Mapping Model (TMM) Analysis Manual* (TR-93-07). Department of Computer Science, Virginia Tech (VPI&SU), Blacksburg, Virginia.

Mayo, K. A. & Hartson, H. R. (1993). *A Methodology for Synthesis of Interface Design Requirements* (TR-93-10). Department of Computer Science, Virginia Tech (VPI&SU), Blacksburg, Virginia.