The GIS User Interface as a Major Economical Factor: A Case Study in Manual Map Digitizing

L'interface utilisateur du SIG comme facteur économique: Le cas de la numérisation de plans

Die GIS Benützerschnittstelle als ökonomischer Faktor: Eine Fallstudie zum Digitalisieren

Werner Kuhn, Peter Haunold, Andrew U. Frank
Technical University Vienna, Department of Geoinformation
Gusshausstrasse 27-29/127, A-1040 Vienna, Austria
Fax: (+43 1) 504-35-35, E-mail: kuhn@geoinfo.tuwien.ac.at

ABSTRACT

Geographic information systems (GIS) have become a widely used technology for managing spatially referenced information in the private and public sectors. The rapid growth of the market for GIS technology and digital spatial data raises some crucial economic issues. While past GIS implementations were primarily concerned with hardware and software capabilities, the focus of attention is now shifting to issues of effective system use by individuals and organizations. Among these issues are the usability of GIS and the cost of data, including their maintenance. This paper presents an analytical investigation of such issues in the context of GIS data acquisition by manual map digitizing at a national surveying and mapping agency.

RESUME

Les systèmes d'information géographique (SIG) sont devenus un outil indispensable pour la gestion de l'information à référence spatiale dans les secteurs privé et public. L'expansion rapide du marché pour cette technologie et pour les données numériques spatiales pose alors d'importantes questions économiques. Tandis qu'au passé, on s'occupaient tout d'abord des capacités des machines et des logiciels, l'attention tourne maintenant vers les questions de l'utilisation productive des systèmes par les individus comme par les organisations. Parmi ces questions sont ceux de la facilité d'utiliser un SIG et ceux du coût des données et de leur mise à jour. Ce rapport présente une étude analytique sur ces questions dans le contexte de l'acquisition des données par une numérisation de plans cadastrales.

ZUSAMMENFASSUNG

Geographische Informationssysteme (GIS) haben sich zu einer Technologie entwickelt, die im privaten und öffentlichen Bereich auf breiter Front zur Verwaltung raumbezogner Information eingesetzt wird. Das schnelle Wachstum des Marktes für GIS und für digitale raumbezogene Daten stellt einige wesentliche wirtschaftliche Fragen. Während in vergangenen GIS Projekten die Leistungsfähigkeit von Hard- und Software im Vordergrund stand, konzentriert sich heute das Interesse auf die Möglichkeiten des produktiven Einsatzes der Systeme durch den Einzelnen und durch ganze Organisationen. Zu diesen Fragen gehören die Benützbarkeit von GIS und die Kosten der Daten mitsamt ihrer Nachführung. Dieser Beitrag stellt eine Untersuchung solcher Fragen vor im Zusammenhang mit der Datenerfassung durch manuelles Digitalisieren in einer nationalen Vermessungsbehörde.

1. INTRODUCTION

Transforming analog map data into digital format by manual digitizing is slow and expensive, but nevertheless widely performed. Manual map digitizing is one of the primary methods of making geographic data accessible to digital processing [Marble and Lauzon 1984]. Studies of digitizing methods in order to find opportunities for optimization are therefore warranted. The work reported here investigates the possibility of applying a model of human-computer interaction, the Keystroke-Level Model, to the analysis and optimization of manual map digitizing tasks.

The Keystroke-Level Model provides a simple method to predict the time expert users need to perform given tasks. The central idea of the model is that the time it takes experts to do a task on an interactive system is determined by the time needed to perform the keystrokes. Therefore, one can determine the execution method used for a task, count the number of keystrokes required, and multiply by the time a single keystroke takes in order to get the total time. Obviously, other elementary operations must be added to the model. These operations are pointing, homing the hand, mental preparation and system response times [Card et al. 1980, Card et al. 1983].

Since the Keystroke-Level Model was introduced in 1980, researchers have applied it to many areas such as text editing, spreadsheets, learning, telephone operator call handling and highly interactive tasks in video games [Card et al. 1980, John 1990, John and Vera 1992, Olson and Nilsen 1988, Singley and Anderson 1988]. For a detailed survey of these and other applications see [Olson and Olson 1990]. The tasks detected in these areas were modeled by the GOMS Model, describing the Goals, Operators, Methods and Selection rules used to solve a task. The Keystroke-Level Model served to analyze and predict execution times of tasks. These times could generally be predicted with high accuracy, suggesting that the model could also be applied to more complex areas. So far, one case study of a user interface for graphical design, VLSI design on the ICARUS computer, has been documented [Card et al. 1983]. There, command execution times have been predicted by the Keystroke-Level Model with about 16% error.

Our case study focused on the transformation of analog map data to digital form at a national mapping agency which establishes a countrywide cadastral database. About 260,000 mapsheets have to be digitized on 150 workstations. 43,000 maps (17%) have been digitized during the first four years of the project. The slow and expensive nature of manual digitizing processes and the lack of reliable, fully automated alternatives necessitate the investigation of the user interface characteristics for digitizing tasks [Frank 1993, Haunold and Kuhn 1993, Kuhn 1990].

The paper presents the methods and tasks for manual map digitizing and an experiment for measuring execution times under production conditions. Two modified keystroke level operators for manual map digitizing are described and measured performance times for digitizing tasks are compared with model predictions. Finally, the use of the Keystroke-Level Model to optimize manual digitizing tasks is demonstrated for a ZOOM/PAN task.

2. MANUAL MAP DIGITIZING TASKS

The goal of manual map digitizing is to build up digital geographic data bases. In the case of the mapping agency in this study, analog cadastral maps are scanned and the scanned images are digitized on screen ("heads-up digitizing"), using a 16-button cursor on a digitizing pad as input device. The result of manually digitizing a part of a cadastral map is shown in figure 1.

A requirement for applying the Keystroke-Level Model to manual map digitizing is the subdivision of digitizing operations into error-free repetitive tasks at the keystroke level. In total, 80 tasks were identified for digitizing parcel boundaries, houses, delineation lines, parcel identifiers, and symbols. This number also includes the tasks for changing and moving map views on screen (zoom and pan) as well as for copying and correcting.

Since the Keystroke-Level Model can only predict execution times, a second requirement is that task acquisition times can be neglected. This reduced the total to 38 tasks which represent some 50% of all possible digitizing tasks, but enables users to perform 90 to 95% of all observed operations. The eliminated tasks were mostly those used for error correction.

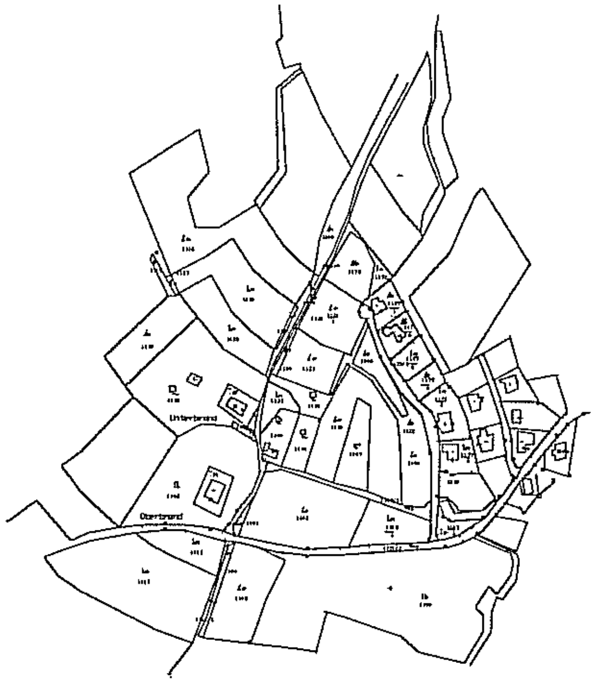


Figure 1: A part of a digitized cadastral map

A third requirement is the subdivision of all tasks into standardized sequences of keystroke-level actions, called unit tasks. Such action sequences were determined for all 38 tasks. The actions (and corresponding model operators) are keystrokes on a typewriter keyboard or button presses on the cursor (**K**), pointing to a target (**P**), mental preparation (**M**), system response times (**R**) and homing hands between keyboard and cursor (**H**).

An example of a digitizing task is *Continuous boundary digitizing* where vertices in a parcel boundary are digitized (see figure 2). For this task the following action sequence was detected:

Pointing to a vertex	**P**
Digitizing the vertex	K [Cursor button 0]

The keystroke-level action sequences determined for each task neglect the mental preparation operator **M**. Adding these operators and defining the resulting sequences as unit tasks was done in the analysis of the experiment described below.

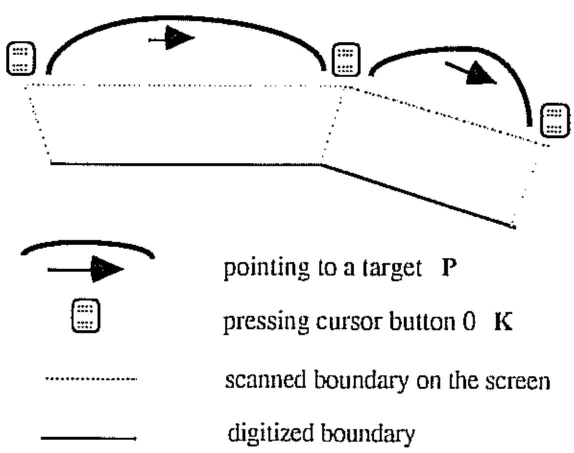


Figure 2: Continuous boundary digitizing

3. EXPERIMENT

An experiment to measure performance times for the 38 tasks was devised and run at the mapping agency under some constraints requiring careful planning.

Planning

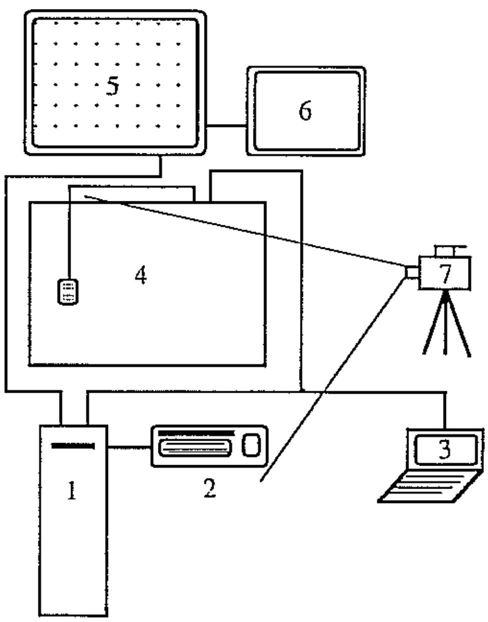
The measurements of performance times were done entirely under production conditions at the mapping agency. Expert users were observed at their regular work places using their normal hard- and software configuration. The agency requested that the experiment should not interfere with the production process. A second constraint on the experiment was to get time stamps at a resolution better than one tenth of a second for each action on the cursor or keyboard.

The agency uses a digitizing program based on AUTOCAD™, installed on a high end personal computer. The input device is a standard high resolution digitizing tablet with a 16 button cursor. Digitizing itself is done on a 21" color monitor.

The idea of only videotaping the actions was dismissed, because the time spent on each action is too short to accurately recognize its start and end. Also, the number of performed actions would be too high for an efficient video analysis. Instead, a program to record the measurement data was written. It had to be set up so that it did not interfere with AUTOCAD™ and the overall digitizing process. The program was installed on a separate laptop computer, connected to the cable between thedigitizing pad and the PC running AUTOCAD™ (see figure 3). The program recorded each signal sent by a button press on the cursor together with a time stamp. Keyboard actions could not be recorded directly. They required additional videotaping for counting keystrokes and measuring homing times.

Running the Experiment

Seven users were observed for two to four hours each while they digitized boundaries, buildings, parcel-ID's and symbols. A total of about 25,000 actions were recorded, corresponding to 10,447 digitizing tasks.



1...Digitizing PC (386er 33 MHz)
2...Keyboard
3...Observation PC (386er 25 MHz)
4...Digitizing Pad
5...Digitizing Monitor
6...Controling Monitor
7...Videocamera

Figure 3: The Hardware Configuration of the Experiment

For the first four users, the video camera was focused on the digitizing pad and the keyboard during the whole observation time. For the other three users, the camera was focused on the screen while they were digitizing lines. This allowed us to determine the times for the system response operator **R** and to examine the times for pointing **P** and mental preparation **M**.

4. ANALYSIS

The analysis of the experimental data started out by verifying whether the existing Keystroke-Level Model operators can be used for manual map digitizing. Two additional operators were introduced. After separating action sequences and defining the unit tasks for manual map digitizing the Keystroke-Level Model was then validated as a whole.

Validation of Existing Operators

According to *Card et al.* their operators provide good results for text editing, operating systems and graphics applications [Card et al. 1980, Card et al. 1983]. In order to test whether the existing operators fit the actions of manual map digitizing the video recordings were used. An overview of the comparison of the existing operators and the operators we found for manual map digitizing is given in table 1.

The time for the pointing operator **P** was estimated as $t_P = 1.16$ sec with standard deviation $\sigma = 0.26$ sec from 50 occurrences. The mean time for the homing operator **H** is $t_H = 0.38$ sec ($\sigma = 0.16$ sec, 50 occurrences). These times show good correspondence to the times given in [Card et al. 1980, Card et al. 1983]. Due to our small numbers of occurrences the times of Card et al. ($t_P = 1.10$ sec and $t_H = 0.40$ sec) will be used to predict task performance times.

Operator	*Card et al.* [sec]	# observed	Result [sec]	σ [sec]
K	0.08-1.20	400	0.25-0.50	0.08
K_C	--	1948	0.28-0.47	0.10
M	1.35	--	1.35	--
P	1.10	50	1.16	0.26
P_S	--	100	0.85	0.19
H	0.40	50	0.38	0.16
R_1	--	120	0.25	0.06
R_2	--	60	0.85	0.18

Table 1: Comparison between existing and map digitizing operators

The time for the mental preparation operator **M** was not tested because of its ill-defined nature. We used the standard heuristic rules [Card et al. 1980, Card et al. 1983] to define unit tasks with this operator. When comparing predicted and measured performance times (see below) these rules and the given time $t_M = 1.35$ were found to provide good matches.

For the system response operator **R** two times were determined for three different actions. The first operator, $\mathbf{R_1}$ was detected both for selecting a new function from the command menu and for building up a ZOOM/PAN window. For both these system times the mean time was calculated to be $t_{R_1} = 0.25$ sec ($\sigma = 0.06$ sec, 60 actions for each). The second operator, $\mathbf{R_2}$, stands for the time it takes the system to close automatically a polygon (e.g., when houses are digitized). This system time is $t_{R_2} = 0.85$ sec ($\sigma = 0.18$ sec), again calculated from 60 measurements.

Definition of New Operators

Two additional operators were introduced for manual map digitizing. The operator $\mathbf{P_S}$ stands for two special pointing actions. The first pointing action is used if a vertex cannot be detected immediately on the scanned original. Users are then extending the pointing line to find the vertex where the line breaks. After detecting it, a second pointing to the vertex is necessary. According to Fitts's Law this pointing time is shorter because of the shorter pointing distance.

The second new pointing action occurs when using a snap function to point at an already digitized vertex. Using the snap function increases the target size. Therefore the pointing time is reduced, again following Fitts's Law. For both new pointing actions a mean performance time of 0.85 sec ($\sigma = 0.19$ sec)was determined from 100 videotaped actions.

The other new operator K_C defines a button press on the 16 button cursor. For the seven observed users different times for a keystroke on the keyboard t_K and a button press on the cursor t_{K_C} were measured. The times range from 0.25 to 0.50 sec, differing by as much as a factor of two for some users. Analyzing all users, the mean values for both operators were estimated with $t_K = 0.40$ sec and $t_{K_C} = 0.39$ sec. Since they are not significantly different, we assume both operators take 0.40 sec ($\sigma = 0.08$ sec).

Analysis of Action Sequences

The goal of this analysis step is to identify unit tasks in observed operations based on possible action sequences and the rules for introducing mental operators. Our analysis revealed multiple sequences of actions for certain tasks.

To show this, the task from figure 2, *Continuous boundary digitizing* as performed by a particular user is used again. In analyzing this task, the videotape and the observation records showed that screen manipulation functions done in between digitizing tasks were affecting their performance times. Thus, the task was split into two new tasks. The first, called T-4, was performed normally, while the second, called T-8, differed in that the user had to reorient himself on the screen after a screen manipulation function (ZOOM/PAN). Therefore, an additional operator **M** for mental preparing was postulated.

The task T-4 corresponded to the task *Continuous boundary digitizing* with the previously given keystroke level operators. Using the standard operator **P** and the measured $t_{K_C} = 0.28$ sec for this particular user, the total performance time for this task is predicted to be:

$$T_{execute} = t_P + t_{K_C} = 1.38 \text{ sec}$$

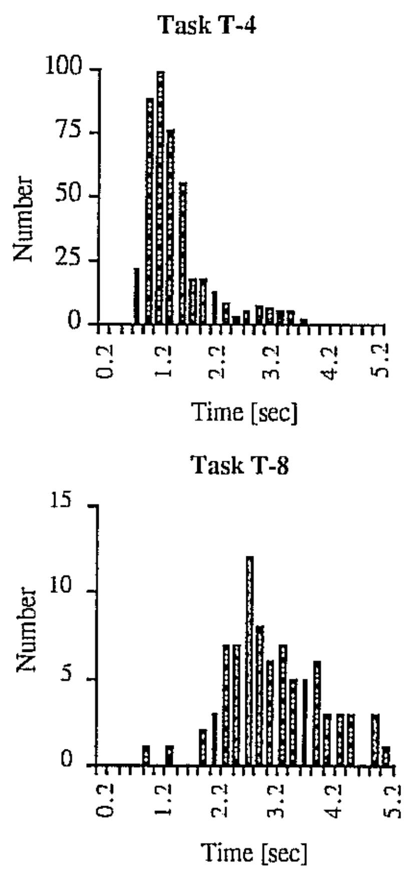


Figure 4: Distribution Function of the two line digitizing Tasks T-4 and T-8

A mean value of 1.25 sec ($\sigma = 0.36$ sec) was calculated from the observations resulting in a difference of 9% between predicted and measured times. Based on our observations and on an analysis of the histogram for this task (see figure 4), we assumed that values larger than 2.4 sec belong to another task where the user needs a second pointing action P_S to digitize a vertex. This assumption will be verified below by statistical tests on task times from all users.

The task T-8 is *Continuous boundary digitizing* plus a mental operator **M** for orientation after a screen manipulation function (ZOOM/PAN). For this task, a modified sequence of actions must be given. The relevant operators from the Keystroke-Level Model are:

Orientation on the screen	**M**
Pointing to a vertex	**P**
Digitizing the vertex	$\mathbf{K_C}$ [Cursor button 0]

The total performance time for this task is predicted to be:

$$T_{execute} = t_M + t_P + t_{K_C} = 2.73\ sec$$

The distribution function for task T-8 is also given in Figure 4. The mean value to perform this task was calculated to 2.98 sec ($\sigma = 0.52$ sec), referring to an approximate normal distribution between 2.0 and 4.0 sec. Thus, the difference between predicted and measured performance times was again 9%.

In order to verify the assumption about multiple action sequences for certain tasks, the three major tasks done by all users were analyzed. These tasks are *Continuous boundary digitizing*, *Continuous boundary digitizing and defining a vertex as control point* and a ZOOM/PAN task.

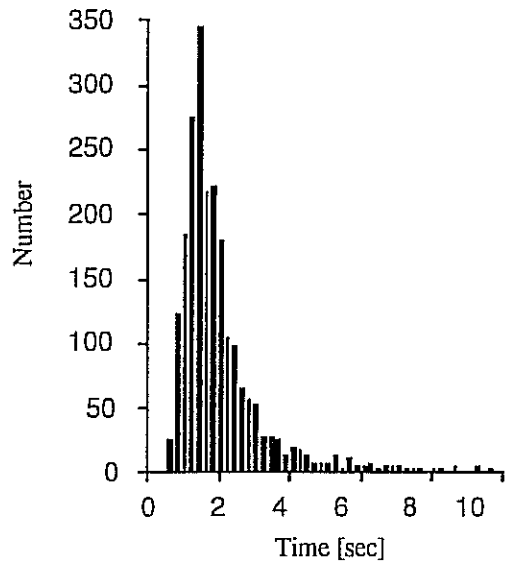


Figure 5: Histogram of the distribution function for the task T-4 *Continous boundary digitizing*

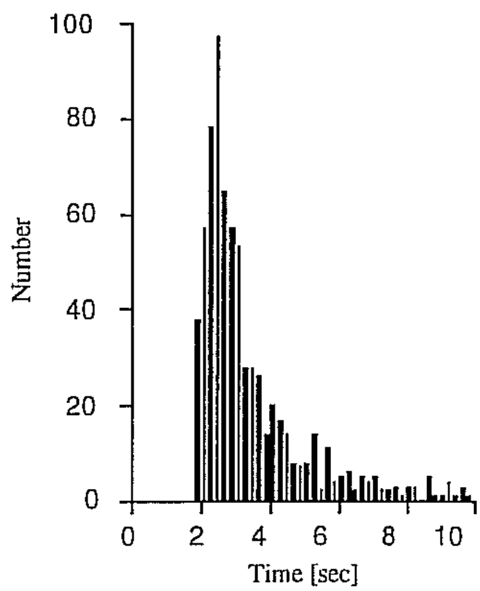


Figure 6: Histogram of reduced data for the task T-4 *Continous boundary digitizing*

For these tasks the distribution functions looked very similar. The histogram of task T-4 is given in figure 5. The first action sequence for this task was modeled as **P**, **K** with a predicted performance time of 1.5 sec. The measured performance time calculated from 1,500 values was also 1.5 sec (σ = 0.3 sec). Assuming a normal distribution between 0.6 and 2.4 sec (corresponding to $\pm 3\sigma$) for this action sequence, and subtracting this effect, the data can be reduced to the distribution function given in figure 6.

When analyzing this new distribution, a second action sequence was detected which corresponded to the first one plus the operator for a second pointing action. Thus, the new task was modeled as **P**, $\mathbf{P_S}$, **K** with a predicted performance time of 2.35 sec. The mean value of 452 measurements between 1.4 and 3.6 sec was 2.54 sec (σ = 0.4 sec). In the same way, yet another action sequence was detected for this task. It corresponds to the second one plus a mental operator **M** for finding a vertex. Thus, the new action sequence was **P**, **M**, $\mathbf{P_S}$, **K**. The match of predicted (3.69 sec) and measured (3.86 sec, σ = 0.5 sec) times was again satisfactory.

The other two major tasks were analyzed in the same way. Each could be subdivided into two different action sequences. The extension of a single unit task distribution was defined to be maximally $\pm 3\sigma$ for each action sequence in agreement with [Card et al. 1980, Card et al. 1983]. As these results confirmed those for the first task, we assume that action sequences can in general be separated with this procedure. Further details on the statistical analyses are contained in [Haunold et al. 1993].

Definition of Unit Tasks and Validation of the Model

Separating different action sequences for the 38 digitizing tasks and defining them as unit tasks resulted in 97 unit tasks. These unit tasks are solving 9,394 out of 10,447 recorded digitizing tasks (90%). About 85% of these unit tasks were performed without errors. For all these error-free unit tasks, mean values and standard deviations of measured performance times were determined. The predicted performance times were calculated using the known keystroke level operators as well as the new operators $\mathbf{P_S}$ for a shorter pointing action and $\mathbf{K_C}$ for a button press on the cursor.

The validation of the Keystroke-Level Model for manual map digitizing is presented comparing predicted and measured performance times for the 30 most frequently performed unit tasks (see table 2). These tasks represent 83% of all defined unit tasks and 63% of all recorded tasks. In this comparison of the data for all users, the operator **K** with t_K = 0.40 sec was used both for a keystroke on the keyboard and for a button press on the cursor. Detailed comparisons for each user can be found in [Haunold et al. 1993].

5. DISCUSSION

Table 2 shows that the predicted times match the measured performance times with an average difference of 5% and a maximum difference of 11%. Comparing these results with those from other keystroke-level analyses published clearly establishes that the Keystroke-Level Model is suitable for a complex graphics application like manual map digitizing.

The discussion of individual results given in table 2 will be limited to a few selected highlights. In particular, interesting cases where mental operators needed to be added following *Card et al.'s* rules are briefly explained based on the observed structure of digitizing operations. This will indicate how detailed application knowledge in conjunction with careful modeling and analysis is able both to validate the Keystroke-Level Model and to support the improvement of GIS user interfaces.

For three boundary digitizing tasks mental operators **M** had to be added because of user decision times. For the task T-3 the operator **M** models the decision where to start digitizing using the snap function for an already digitized vertex. The **M** in task T-5 stands for deciding which vertex is digitized next if there are several possibilities (e.g., if boundaries are crossing). In task T-14 **M** stands for deciding whether digitizing should be ended or not.

Task	Function	Operator sequence	Number	T_m.[sec]	T_p[sec]	D[%]
T-1	Start boundary digitizing with command selection	M,P,K,M,P,K	10	5.26 ± 0.36	5.70	-8
T-2	Start boundary digitizing using a snap function	2K,P_S,K	164	2.28 ± 0.32	2.05	11
T-3'	Start boundary digitizing using a snap function	M,2K,P_S,K	181	3.08 ± 0.58	3.40	-9
T-4	Continuous boundary digitizing	P,K	1621	1.57 ± 0.34	1.50	5
T-5'	Continuous boundary digitizing	M,P,K	103	3.11 ± 0.44	2.85	9
T-6	Continuous boundary digitizing using a second pointing action	P,P_S,K	185	2.35 ± 0.25	2.35	0
T-7	Continuous boundary digitizing using a second pointing action	P,M,P_S,K	147	3.65 ± 0.61	3.70	-1
T-8*	Continuous boundary digitizing	M,P,K	463	3.03 ± 0.54	2.85	6
T-9*'	Continuous boundary digitizing using a second pointing action	M,P,M,P_S,K	163	5.06 ± 0.61	5.05	0
T-10	Continuous boundary digitizing and defining a vertex as control point	P,2K	676	1.97 ± 0.40	1.90	4
T-11*	Continuous boundary digitizing and defining a vertex as control point	M,P,2K	128	3.22 ± 0.44	3.25	-1
T-12	End boundary digitizing	P,3K	147	2.41 ± 0.48	2.30	5
T-13	End boundary digitizing using a snap function	K,P_S,3K	252	2.43 ± 0.35	2.45	-1
T-14'	End boundary digitizing using a snap function	K,P_S,M,3K	105	3.65 ± 0.44	3.80	-4
T-15	Continuous house digitizing	P,K	292	1.55 ± 0.32	1.50	3
T-16	Continuous house digitizing using a snap function	K,P_S,K	105	1.69 ± 0.35	1.65	2
T-17	End house digitizing and close polygon	4K,R_2	16	2.56 ± 0.30	2.45	4
T-18	End house digitizing using a snap function and close polygon	K,P_S,4K,R_2	26	3.39 ± 0.33	3.70	-8
T-19	Continuos line digitizing for land use	P_S,K	180	1.18 ± 0.32	1.25	-6
T-20	Start symbol digitizing with command selection	M,P,K,M,P_S,K	94	5.80 ± 1.03	5.45	6
T-21	Continuous symbol digitizing	P_S,2K	245	1.68 ± 0.36	1.65	2
T-22	Continuous symbol digitizing	P_S,K	80	1.33 ± 0.24	1.25	6
T-23	ZOOM/PAN task	K,R_1,P,K,R_1	712	2.64 ± 0.35	2.40	10
T-24'	ZOOM/PAN task	M,K,R_1,P,K,R_1	265	3.70 ± 0.54	3.75	-1
T-25	ZOOM/PAN task	K,R_1,P,K,P_S,K,R_1	60	4.06 ± 0.54	3.65	11
T-26'	ZOOM/PAN task	M,K,M,P,K,P_S,K,R_1	58	6.54 ± 0.95	6.10	7
T-27	Start parcel-id digitizing with command selection	M,P,K,R_1,H,M,5K,H,P_S,K	10	9.00 ± 0.46	8.50	6
T-28	Continuous parcel-id digitizing	M,K,5K,P_S,K	38	5.08 ± 0.37	5.00	2
T-29	Continuous parcel-id digitizing	M,K,H,5K,H,P_S,K	30	6.43 ± 0.74	5.80	11
T-30	Continuous parcel-id digitizing	M,K,H,4K,H,P_S,K	66	5.55 ± 0.79	5.40	3

Table 2: Comparison of measured and predicted performance times.
A * after the task number means that this task is performed after a ZOOM/PAN task.
A ' after the task number means that a mental operator is added.

Task T-19 describes the digitizing of lines other than boundaries. These lines only have delineation function (e.g., for land use areas), whereas parcel boundaries have legal and economic functions. Therefore, the lines are digitized less accurately, and this can be modeled using the second pointing operator $\mathbf{P_S}$.

Some interesting observations can be made for ZOOM/PAN tasks. The tasks T-23 and T-24 are used for a simple zooming task. Both tasks have the same results but some users need mental preparation **M** to perform them (task T-24). Analyzing the sequence of actions revealed the use of a complicated method which explains the need for mental preparation: A window showing zoom and pan commands is opened, then the zoom command is selected, reducing or enlarging the view centered on the screen. Tasks T-25 and T-26 are also zooming tasks but users can select where they want to zoom. Some users need mental preparation **M** again (task T-26).

To show how tasks can be optimized using the Keystroke-Level Model, we analyze a modified ZOOM/PAN task. In task T-23 the command to open a window is pressing the button 9 on the cursor (**K**). Then, a system time $\mathbf{R_1}$ is required to open the window. Choosing the zoom or pan command is done by pointing **P** and another button press **K**. To redraw the modified view centered on the screen, a system time $\mathbf{R_1}$ is needed again. The performance time for this task is thus predicted with 2.40 sec (see table 1).

An optimized sequence would need a button press **K** to select the zoom or pan command and the system time $\mathbf{R_1}$ for redrawing. Some of the 16 buttons would have to be programmed for choosing these commands. The resulting performance time is predicted to be 0.65 sec. Since this particular ZOOM/PAN task was found to be used during roughly 10% of the routine digitizing time, an alternative design reducing the performance time by as much as 3/4 would offer significant optimization potential.

6. CONCLUSIONS

This study investigated routine tasks in manual map digitizing. The experimental data were collected in a major national digital mapping project and analyzed with the Keystroke-Level Model. The main goal was to establish the suitability of this model for manual digitizing operations as a first step toward using the model for further optimizing these costly GIS operations.

Our results indicate that the Keystroke-Level Model is clearly applicable to manual map digitizing. This holds true under the usual restrictions to error-free, routinely performed tasks and with the definition of two additional operators for special pointing and keying actions. We found an average of 5% and a maximum of 11% difference between predicted and measured task performance times.

Multiple intertwined action sequences for certain tasks could be successfully separated combining application knowledge about task semantics with the conventional rules for placing mental operators. Furthermore, considerable optimization potential was found and demonstrated for the case of an intensely used zoom and pan function for which performance time can theoretically be reduced by 73%.

The study produced a wealth of observation data (more than 10'000 actions[1]) and suggests a broad range of further analyses. While our validation of the model focused on average performance times for all observed users, studies of individual differences and their patterns seem to be indicated. Also, recent extensions of the model to accommodate parallel actions [Olson and Nilsen 1988] could be applied, for example, to two-handed input from cursor and keyboard.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution from Intergraph Corporation and the support from the Austrian surveying and mapping agency.

REFERENCES

Card, S.K., Moran, T.P., & Newell, A., *The Keystroke-Level Model for User Performance Time with Interactive Systems*. ACM Communications, 23(7), 1980, pp. 396-410.

Card, S.K., Moran, T.P., & Newell, A., *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

Frank, A., The use of Geographical Information System: The user interface is the system, in Medickij-Scott, D. & Hearnshaw, H. (eds.) *Human Factors in GIS*. Belhaven Press, London, 1993, pp. 3-14.

Haunold, P. & Kuhn, W., A Keystroke Level Analysis of Manual Map Digitizing, in Frank, A.U. & Campari, I. (eds.) *Spatial Information Theory*. Springer, Lecture Notes in Computer Science, vol. 716.

Haunold, P., Kuhn, W., & Frank, A.U., *A keystroke level analysis of manual map digitizing (in German)*. Technical University Vienna, Dept. of Geoinformation, 1993.

John, B.E., Extensions of GOMS Analyses to Expert Performance Requiring Perception of Dynamic Visual and Auditory Information, *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*. 1990, pp. 107-115.

John, B.E. & Vera, A.H., A GOMS Analysis of a Graphic, Machine-Paced, Highly Interactive Task, *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*. 1992, pp. 251-258.

Kuhn, W., Editing Spatial Relations. In proceedings of *Fourth International Symposium on Spatial Data Handling* (Zurich, Switzerland; July 1990) 1990, pp. 423-432.

Marble, D.F., Lauzon, J.P., & McGranaghan, M., Development of a Conceptual Model of the Manual Digitizing Process. In proceedings of *International Symposium on Spatial Data Handing* (Zurich, Switzerland) Geographisches Institut, Abteilung Kartographie/EDV, 1984, pp. 146-171.

Olson, J.R. & Nilsen, E., *Analysis of the cognition involved in spreadsheet software interaction*. Human-Computer Interaction, 3, 1988, pp. 309-350.

Olson, J.R. & Olson, G.M., *The growth of cognitive modeling in human-computer interaction since GOMS*. Human-Computer Interaction, 5, 1990, pp. 221-265.

Singley, M.K. & Anderson, J.R., *A Keystroke Analysis of Learning and Transfer in Text Editing*. Human-Computer Interaction, 3(3), 1988, pp. 223-274.

---

[1] Our observation data are available in spreadsheet format (tab separated), together with a user guide, by anonymous ftp from ftp.tuwien.ac.at.