A survey on user expectations for interface builders

M.C. Desmarais, C. Hayne, S. Jagannath, and R. Keller

Centre de recherche informatique de Montréal
1801 ave. McGill College, bureau 800, Montréal, Québec, Canada H3A 2N4
Tel: 514-398-1234 Fax: 514-398-1244 E-mail: desmarais@crim.ca

ABSTRACT

This study provides many insights into the features that users look for in interface building tools, as well as those that can hinder their use. The results suggest that users are willing to pay a high price for a reliable tool that will combine both fast prototyping and fully operational target interfaces and at the same time provide high functionality. Yet users want a tool that is easy to use and to learn. This is a great challenge for developers of interface builders since there is currently a compromise to make between tools that allow fast and easy prototyping, and toolkits which provide high functionality and good execution speed.

Keywords: GUI tools, survey, evaluation.

INTRODUCTION

Graphical user interface building tools have flourished in the last five to ten years. They range from libraries of routines, like the early MacIntosh toolbox or the OSF Motif widgets, to more recent direct manipulation environments, like VisualBasic or UIM/X, that allow the interactive assembly of interface components (see Myers, 1992, and Hartson & Hix, 1989 for taxonomies and a review of such tools).

Due to the increased importance this technology has had in recent times, we conducted a survey among development teams of interactive applications on their use of interface building tools. We report here some results from the survey that relate to the features that are a hindrance to the use of interface building tools as well as features that are most desired by development teams. This constitutes a complement to the survey conducted by Myers and Rosson (1992) on user interface programming in which they report informal and qualitative results on the evaluation of interface building tools. See also Hix, 1991, for related work on the evaluation of interface builders.

METHODOLOGY AND SAMPLE CHARACTERISTICS

The survey was conducted in September 1993 and consisted of a questionnaire composed of 19 questions. The questionnaire was administered to one member from each of 56 software development teams in 40 different organizations, situated in the provinces of Quebec and Ontario. The survey's personalized approach resulted in a high response rate of 100%. These organizations were composed of 19 small medium size companies (25 respondents), 19 large organizations and governmental bodies (26 respondents), and 5 university/research teams (5 respondents).

Seventy six percent of the respondents had used at least one interface builder, the others knew them from a managerial perspective. Most teams were operating within Microsoft Windows and the X-Window environments (87% and 56% respectively) while OS/2 and the Macintosh environments were each used by about a third of the teams. Twenty percent were operating within a mainframe environment (all but one of which were large organizations). The languages most used were C and C++ (79% and 61% respectively).

USAGE OF TOOLS

Usage of interface building tools was widespread in our sample as only two of the 56 respondents did not report any tool used. This percentage is greater than the one obtained by Myers and Rosson (1992), probably because we also included here occasional use and the use for prototyping purpose only, whereas Myers and Rosson's study focused upon one particular software development project by respondent.

Thirty-one respondents (58%) used only interactive GUI development tools, whereas 5 respondents (9%) used only non-interactive tools. The remaining 33% used both interactive and non-interactive tools.

The interactive tool most widely used was Visual Basic (41% used it—22 out of 54 respondents who used tools). Hypercard came second with 30%. UIM/X (28%), Visual C++ (28%), and XVT (26%) followed. HyperCard was the tool most preferred for prototyping (all 16 users of HyperCard), but only 31% of them built operational systems with it (5/16). Visual Basic was used for prototyping by 86% (19/22) but less than half as many built operational systems with it (9/22). UIM/X and Visual C++ were the preferred tools for building operational systems with around half of their usage devoted to this purpose (9/15 and 7/15 respectively).

FEATURES DESIRED IN INTERFACE BUILDING TOOLS

Table **??** contains results for the 53 respondents to the question concerning the importance of features offered by user interface development tools. The features are sorted in order of decreasing importance. The first section of the table consists of features of the development tool itself whereas

Short paper presented at the CHI 94 conference.
©1994 ACM

Table 1: Importance of interface builder's features

Features of interface devlop. tool	++	+	–
1. Performance of tool (bug free and reliable)	43	10	0
2. Tool's rapid prototyping capabilities	38	13	2
3. Complete system development (not just prototypes)	36	13	4
4. Content and quality of documentation	32	17	4
5. Modular development	32	20	0
6. Ease of using the tool	31	22	0
7. Reusability of tool outputs	30	20	1
8. Speed of execution of tool	29	24	0
9. Easy linkage to other libraries and tools	26	21	5
10. Availability of classes / libraries for use	25	22	5
11. Error detection/correction	22	27	4
12. Technical support offered for tool	21	30	2
13. Tool's ability to enforce interface standards	21	22	6
14. On-line help in the tool	20	21	12
15. Ease of learning and installation	19	29	5
16. Portability aspects of tool	15	27	11
17. Customizability of tool	11	28	14
18. Code generation for multiple languages	11	15	26
19. Cost of tool	8	38	6
20. Project management capabilities	6	19	22
Features of target interface	++	+	–
1. Speed of execution	33	17	2
2. Run-time licensing conditions	32	10	6
3. Provision of on-line help	27	21	4
4. 2D graphical interfaces	24	21	5
5. Object-oriented development	23	21	7
6. Database linkage	22	20	10
7. Customizability of interfaces	22	25	6
8. Cross-platform portability	21	23	8
9. Multimedia applications	16	15	18
10. Hypertext applications	15	10	21
11. Graphical animation	10	19	22
12. 3D graphical interfaces	6	20	22
13. Speech input/output interfaces	5	15	29

++ : Very important + : Fairly important - : Not important

the second section contains features of the target interface created with a tool. The choice of these features was inspired by their existence in currently available commercial tools.

Very high on the list were two apparently opposed features: The tool's rapid prototyping capabilities (2) and its ability to be used for operational systems (3). Combined with the tool's ease of use (6), those features constitute a true challenge to integrate.

General software quality of the tools was also considered very important by respondents. Features rated as highly desirable were: bug free and reliable performance (1), good documentation (4), and execution speed (8).

Modularity and reusability of tools were also factors that are high on the list (5,7), suggesting that users look for software engineering advantages in using interface building tools.

Interestingly enough, the cost of the tool was rated second lowest for "Very important" but it is highest on the "Fairly important" scale, which suggests that users are willing to pay a high price for a tool that meets their expectations, even though the cost will be a deciding factor.

Table 2: Hindrances to using interface builders

Hindrance	Res.
1. Lack of knowledge about such tools	25
2. Lack of time to investigate the tools	25
3. The tools do not have the functionality we need	23
4. Lack of experienced developers on the market	20
5. Fear of adapting a tool which becomes defunct	20
6. Lack of standardization	19
7. Extra cost involved in learning the tool	17
8. Tools are too expensive	16
9. Fear of adapting a non-standard approach	15
10. Complexity of usage	13
11. Our applications are too special purpose	11
12. Not applicable (There is no reason hindering their use)	4

Res.: Number of respondents out of a total of 56.

Turning to the features that are sought for the target interfaces, the two most important features are speed of execution and run-time licensing conditions, which attest to the concern for operational interfaces. The third most important feature, namely the ability to integrate on-line help, attests to the concern for the usability of the target interface.

HINDRANCE TO USING INTERFACE BUILDING TOOLS

We also asked the respondents to provide reasons that would most likely hinder the widespread use of user interface development tools in their organizations. The results are compiled in table **??**.

Very high on the scale are reasons that relate to the expertise around interface builders. Lack of knowledge (1), lack of time to investigate the tools (2), and lack of experienced developers (4) are all factors that reveal this problem. This is not surprising given the proliferation of these tools and the complexity of the task of analysing them in details, especially given the wide range of approaches they offer to building interfaces (eg. source code generation, run-time proprietary window management, compile-time libraries, etc.).

The second set of most important reasons against the use of interface builders relate to the problem of standardization (6). Users do not want to be tied to a tool (5) or adopt a non-standard approach (9).

Finally, it is worth noting that lack of functionality (3) and the cost of learning the tools (7) are also high on the list.

REFERENCES

1. H. R. Hartson and D. Hix. Human-computer interface development: Concepts and systems for its management. *ACM Computing Surverys*, 21(1):5–92, 1989.
2. D. Hix and R. S. Schulman. Human-computer interface development tools: A methodology for their evaluation. *Commun. ACM*, 34(3):74–87, March 1991.
3. B. A. Myers. State of the art on user interface software tools. *Advances in Human Computer Interaction*, 4, 1992.
4. B. A. Myers and B. M. Rosson. Survey on user interface programming. In *Proceedings of CHI'92*, pages 195–202, New York, 1992. ACM.