first two phases of this proposed paradigm are similar to the spiral model, except that Web documents have different focuses as discussed earlier. Since Web documents are to deliver information, the content is more important than the feasibility analysis of a system, as it is in a software system. We also incorporate testing in the process of maintenance, since Web documents should be tested for obsolete URLs and missing tag objects in its maintenance. Instead of using risk analysis after testing, Web document assessment should be used in each and every cycle of the spiral model. The result of content assessment and testing decides whether the Web document designer should deliver a version of the document, or continue to go through another iteration in the spiral model.

The development activities in the three phases (i.e., script, implementation, and testing/maintenance) as well as Web document assessment are based on the status of a document repository. With a centralized repository, this paradigm enforces the consistent changes in each step of the development cycle. Starting from a requirement, the developer designs a Web script. This script can be a text description of the requirement of a home page, with optional multimedia resources (e.g. picture, sound, or video) as on-line references. Alternatively, the script can be presented in a spoken language and recorded as audio records. Whether the script is recorded in text or voice, the implementation of the Web script is produced either automatically or semi-automatically. The physical testing can be conducted afterwards. The Web document will be delivered after a number of iterations in the spiral model. If the content of this Web document is suitable, a version of this document is delivered. However, the document should be constantly tested, retested and updated. The development life cycle of a Web document can be considered at three different levels: intra-directory level, intra-station level, and inter-station level. Documents at each phase are designed depending on the range of references or URLs. For instance, a document script may refer to a multimedia resource file within the local station, or another Web page in a remote site. The implementation will use different ranges of URLs with respect to the script. The testing scope of the URLs can thus be decided afterwards.

This Web document development paradigm is yet another example of applying software engineering methods to the development of multimedia systems.

8. Multimedia Software Project Effort Measurement

In Sec. 1 of this chapter, we discussed how multimedia technology is utilized in software engineering project management. In this section, we discuss how to apply software engineering technology in multimedia project management, and address the issue of multimedia software project effort measurement.

Why multimedia software project effort measurement is an issue? Can't multimedia projects be treated in the same way as other types of software projects? To answer this question, we note that there have been substantial efforts in measuring traditional transaction processing and process control systems, but relatively

little effort in determining and evaluating multimedia effort measurements. In fact multimedia software is unique in its emphasis on contents and the need for storyboarding early in the development cycle. Because of that, we not only need tools and techniques specific to multimedia, but also should emphasize different aspects in software project management. For one, managers from diverse background are needed for such projects, to form cross-disciplinary teams. For another, due to the inherently different nature of multimedia systems, the direct application of existing models and measures may not meet multimedia project management needs.

A preliminary empirical study to develop an algorithmic mapping to "effort" using product characteristics appropriate to multimedia software systems other than LOC (Line of Codes) or external files is reported in [12]. The development effort is regarded as a function of (1) building the system content — each media form might have a different impact on development effort — including file name, media type, creation effort, digitizing effort, editing effort, and (2) authoring the system — a screen that incorporates a greater number of objects and events would take proportionally greater effort to develop. The empirical study on projects developed by senior-level students and delivered in two stages: prototype and final delivery. The students are information science students focussed on programming and design students concentrated on content and interface development. Forty-five observations were recorded in media component development, but no correlation between development effort and either of the component variables — media type and media status. Since data sets were small, generalizations are not possible from this study.

There is a lack of industry driven determination of important attributes. Moreover, data collection needs to be exploratory and student projects are not representative of industry developments. Thus, there is a need to develop an industry-based metrics framework in order to determine system and component characteristics considered influential in multimedia software system development efforts. The Goal/Question/Metric (GQM) model shown in Fig. 6 is one such framework.

The key issues considered to reflect industry perspective include:

- Development Tools: Authoring tools and other more complex tools offering visually based high productivity development languages or high level scripting languages.
- Delivery Platform: Platform dictates media format and optimization considerations, for example, CD-I must conform to PAL or NTSC.
- Content Development: Media content development is labor intensive, but difficult to quantify. It depends on media mix, the number of components, and complexity of each component. Artistic considerations also play a role in satisfactory completion of project.
- Organizational Capability: Indirect effects such as size, personnel mix, etc.
- Personnel: Wide variety of skills is required in successful multimedia software development. The core members are a producer, programmer(s), and graphic artist(s).

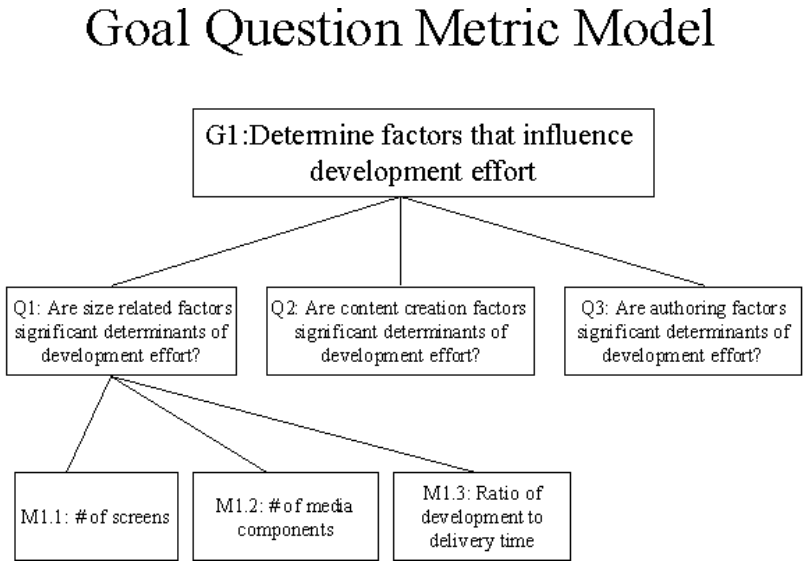

Fig. 6. The goal/question/metric model (a small portion is shown for illustrative purpose).

To perform preliminary verification of the above framework, a pilot study was conducted [21]. Structured interviews were conducted with three multimedia development organizations. Results suggest the type of project almost entirely determines development environment and workload. Results of pilot study were combined with components of GQM framework to develop a postal survey. The focus is to determine factors that influence development efforts. It was found that 50% of respondents use no formal methodology. None of the traditional software metrics is used. Rather, experience from previous projects is used. Project tracking is performed at very high level, indicating immature project management. Staff experience and project size are obviously important factors.

To summarize, (1) traditional software metrics such as COCOMO and FPA are usually not utilized in multimedia software development; (2) pilot study and questionnaire reinforce the non-formal approach to multimedia software development; (3) although there are attempts to formalize the approach to software development, there is uncertainty on how to achieve this. In other words, multimedia software engineering as a scientific discipline is still evolving, making it an exciting research area to explore.

References

1. J. F. Allen, "Time and time again: The many ways to represent time", *International Journal of Intelligent Systems* **9**, no. 3 (June 1991) 413–428.
2. T. Arndt, A. Cafiero and A. Guercio, "Multimedia languages for teleaction objects", *Proceedings of 1997 IEEE Symposium on Visual Languages* (September 1997) 318–327.

3. T. Arndt, A. Cafiero and A. Guercio, "Symbol relation grammars for teleaction objects", Technical Report, Dipartimento di Informatica ed Applicazioni, University of Salerno (1997).
4. T. Berners-Lee, R. Calliau, J. F. Groff and B. Pollermann, "World-wide web: The information universe", *Electronic Networking* **2**, no. 1 (Spring 1992) 52–58.
5. S. K. Chang, "Towards a theory of active index" *Journal of Visual Languages and Computing* **6** (1995) 101–118.
6. H. J. Chang, T. Y. Hou, A. Hsu and S. K. Chang, "The management and application of tele-action objects", *ACM Multimedia Systems Journal* **3**, no. 5–6 (Springer Verlag, 1995) 204–216.
7. S. K. Chang, "Extending visual languages for multimedia", *IEEE Multimedia* **3** (1996) 18–26.
8. S.-K. Chang, *Multimedia Software Engineering* (Kluwer Academic Publishers, 2000).
9. S. K. Chang, E. Hassanein and C. Y. Hsieh, "A multimedia micro-university", *IEEE Multimedia Magazine* **5**, no. 3 (July-September 1998) 60–68.
10. T. Chiueh and W. Wu, "Variorum: Multimedia-based Program Documentation System", Technical Report, CS Department, SUNY, Stony Brook (1998).
11. A. Dattolo and V. Lois, "Active distributed framework for adaptive hypermedia", *International Journal of Human Computer Studies* **46**, no. 5 (May 1997) 605–626.
12. T. Fletcher, S. G. MacDonell and W. B. L. Wong, "Early experiences in measuring multimedia systems development effort" (1998).
13. F. Ferrucci, G. Pacini, G. Satta, M. I. Sessa, G. Tortora, M. Tucci and G. Vitiello, "Symbol relation grammars: A formalism for graphical languages", *Information and Computation* **131**, no. 1 (1996) 1–46.
14. A. Garrido, G. Rossi and D. Schwabe, "Pattern systems for hypermedia", *Proceedings of the PloP'97* (1997).
15. F. Halasz, "Reflections on notecards: Seven issues for the next generation of hypermedia systems", *CACM* **31**, no. 7 (1995) 836–855.
16. M. Hirakawa, "Do software engineers like multimedia?" ICMCS'99, Firenze, Italy (June 1999).
17. T. Hochin, M. Harada, M. Nakata and T. Tsuji, "Virtual multimedia objects as the platform of customizing of multimedia data", *Proceedings the International Workshop on Multimedia Software Engineering*, Kyoto, Japan (April 1998) 20–27.
18. T. Isakowitz, E. A. Stohr and P. Balasubramanian, "RMM: A methodology for structured hypermedia design", *CACM* **38**, no. 8 (August 1995) 34–44.
19. T. D. C. Little and A. Ghafoor, "Interval-based temporal models for time-dependent multimedia data", *IEEE Transactions on Data and Knowledge Engineering* **5**, no. 4 (1993) 551–563.
20. D. Lyardet, G. Rossi and D. Schwabe, "Using design patterns in educational multimedia applications", *Proceedings of the ED-MEDIA'98*, Friburg (June 1998).
21. S. G. MacDonell and T. Fletcher, "Industry practices in project management", *Proceedings of the Metric'98* (IEEE CS Press).
22. S. Sauer and G. Engels, "Extending UML for modelling of multimedia applications", *Proceedings of IEEE Symposium on Visual Languages* (Tokyo, 1999) 80–87.
23. T. K. Shih, S.-K. Chang and P. Shih, "A web document development paradigm and its supporting environment", *Proceedings of the Sixth International Conference on Distributed Multimedia Systems* (*DMS'99*), University of Aizu, Japan (July, 1999) 26–30.

24. J. P. Tsai, "Knowledge-based software architecture", *IEEE Transactions on Knowledge and Data Engineering* **11**, no. 1 (January 1999).
25. J. C. Wild, K. Maly and L. F. Liu, "Decision-based software development", *Journal of Software Maintenance* **3**, no. 1 (1991).
26. J. C. Wild *et al.*, "Project management using hypermedia CASE tools", Technical Report, Department of Computer Science, Old Dominion University (1998).