

Finding method in the madness **Methodologies for developing knowledge-based systems**

Nancy Martin and Howell Huws, Coopers & Lybrand

For a number of years now, developers of conventional systems have tried to standardize as many aspects of development as possible, whether by automating analysis and design or by using standard process models to guide inexperienced practitioners. The appearance and acceptance of structured development methodologies was one of the dominant trends in conventional software development in the 1970s and 1980s.

Standard methodologies have many benefits. They allow more flexibility when assigning people to projects, since the method of working is known to everyone. Standardized documentation formats make it easier for reviewers to provide quality control. In many cases, linking standard techniques to powerful automated tools has increased productivity significantly.

Some people, though, view standard methodologies as too static and inflexible. Indeed, one of the things that first attracted many developers to knowledge-based systems was their lack of a methodology and the associated overhead. Early on, there was no general agreement on the components of a development methodology for knowledge-based systems — not on the process model, the content of the “deliverables,” the techniques, or the tools to support them. Also, since early development efforts were stand-alone affairs with very small project teams — typically one expert and a project manager working with one or two knowledge engineers — knowledge-based-systems developers did not see the need for a methodology at all. But as more companies realize the benefits of knowledge-based systems, they are turning to larger, integrated solutions that require more complex development teams with several experts and user representatives.

Strategically, knowledge-based systems are now central to many companies' information-systems portfolios. This is a sign of the commercial maturity of knowledge-based systems, but this success brings many of the same problems faced by developers of conventional

As more companies realize the benefits of knowledge-based systems, they are turning to larger, integrated solutions that require more complex development teams with several experts and user representatives.

systems in the 1970s. For this reason, many people are trying to introduce more structure and standards into the development of knowledge-based systems.

Methodologies in the marketplace

A panel recently met at the Aion User Conference in Monterey, California. Each of the “Big Six” accounting firms — Andersen Consulting, Coopers & Lybrand, Deloitte-Touche, Ernst & Young, KPMG Peat Marwick, and Price Waterhouse — was asked to make a presentation, addressing a list (produced initially by Coopers & Lybrand) of criteria for judging the perceived value and usefulness of development methodologies for knowledge-based systems. (These firms have used conventional software-development methodologies for years.) The criteria were

- *Origin* — Is the methodology conventional and used without adaptation, adapted from an existing conventional methodology, or totally new?
- *Relationship with conventional methodology* — What is the relationship between the development methodologies for knowledge-based systems and for conventional systems?
- *Business-oriented approach* — Does the methodology encompass business considerations (budget cycle, team cohesion, window of effective implementation, and so on) as well as technical ones?
- *Task structure* — How are the tasks defined and grouped?
- *Deliverables* — What key deliverables does the project require? How are they linked to the tasks and each other? What help is provided for writing documentation?
- *Process model* — What model is used for managing projects, and what is its relationship to the “waterfall” model and other paradigms? Does the methodology allow software to be developed iteratively? How is the iterative development of deliverables and system documentation managed? How are iterative projects estimated? How is project control maintained?
- *Training* — What role does training play in the methodology? What training is now available?
- *Integration with CASE tools* — Can (or should) CASE tools be used to develop knowledge-based systems?
- *Knowledge modeling* — How and when is the appropriate knowledge model selected?
- *Knowledge acquisition* — How is knowledge acquisition defined and used?
- *Integration with conventional systems* — How does the methodology address

integration of a knowledge-based system with conventional systems? What features facilitate the transfer (to conventional-systems developers) of development techniques or technologies for knowledge-based systems?

- *Choosing applications* — How should applications be selected and scoped?
- *Controlling time and costs* — How are a project's time and costs controlled, and how is the degree of project completeness measured?
- *Choosing a project team* — What team roles are defined, and how should the team be selected?
- *Project management* — What is the project manager's role?
- *Change control* — How is change controlled, and who controls it?

Five of the six firms described their methodologies, based on these issues. Ernst & Young, which does not have a methodology for knowledge-based systems in the US, presented a list of desired features for the next generation of methodologies. Given the limited presentation time, it was impossible to give each point the treatment it would warrant in a full-scale evaluation. However, common threads did emerge from the presentations.

Most of the firms have used their conventional methodology as a framework for their knowledge-based-systems methodology. This provides common terminology, project structure, and documentation principles that the conventional-system developers already know. This commonality also helps development efforts to integrate knowledge-based systems with conventional systems.

All the firms (except Andersen) included a detailed iterative life cycle that combined management needs (to monitor, motivate, and control) with the needs of knowledge engineers (the flexibility and creativity required to explore alternatives). All six firms also recognized that a complete methodology should include technical guidance on knowledge representation and acquisition. In particular, several firms mentioned the use of knowledge acquisition and design support (KADS, a knowledge-modeling methodology). Deloitte-Touche's firm in the United Kingdom was part of the KADS development effort, and Andersen, Price

Waterhouse, and Coopers & Lybrand all exploited the public-domain "interpretation models" made available as a result of KADS work in Europe.

Each methodology has strengths and weaknesses, and no vendor stood out by virtue of completeness or authority. It is difficult to be sure whether differences between the methodologies are differences of omission or of emphasis.

KPMG Peat Marwick emphasized details of the advice provided for each development activity and showed how

Knowledge-based systems are now central to many companies' information-systems portfolios, but this success brings many of the same problems faced by developers of conventional systems in the 1970s.

certain activities require different approaches when developing knowledge-based systems. The firm said its methodology comprises a "complete business solution" approach, of which a knowledge-based system is just a part. The company is focusing on knowledge acquisition and representation as the area of principal differences from the conventional process model: if these are found to be sufficient, then they might define expert systems as a separate process model.

Andersen emphasized the completeness and depth of model support, which is required in a methodology that views the selected knowledge model as the principal source of design and knowledge acquisition. The company also highlighted its tool support for the entire development process, whether for computer-aided software engineering (via the Foundation tool) or for project management (via Manage/1).

Deloitte-Touche briefly introduced its source methodologies (4Front and KADS I), focusing on their strengths and lineage. The firm promised that the methodologies would be effectively integrated with each

other soon, and that 4Front would continue to map to KADS as it developed. The company also pointed out that 4Front emphasizes process management rather than task management.

Coopers & Lybrand was the only firm to present an overall life cycle, from project conception, through different stages of maturity in understanding the problem and its implementation, until the system is handed over to maintenance. At the end of each stage, a functioning system with full documentation is available for review and, if appropriate, deployment. Summit-DK requires a separate design phase to ensure that the system can be safely enhanced and expanded during iterations. In its full knowledge-based-system flavor, this is the only methodology now available for sale.

Price Waterhouse's presentation of its System Management Methodology: System Development for Knowledge-Based Systems (or SMM/SDK) spotlighted the firm's detailed approach to knowledge acquisition. The company says its support in this area (which does appear considerable) is so comprehensive that it warrants the label "methodology" itself. (I would have liked so clear and detailed an explanation of the place of iterative development in SMM/SDK.)

Developers are clearly coming to recognize the limitations of their methodologies in dealing with innovative technologies beyond database-and-Cobol applications. While the presenters referred to much valuable research that has already been done, we have likely not heard the end of how these methodologies will incorporate such research for commercial use. However, even using these methodologies in their present state provides real benefits and removes some of the risks of developing knowledge-based systems, increasing their likely commercial acceptance.

Nancy Martin is a partner in charge of the Advanced Technology Group at Coopers & Lybrand, and a member of IEEE Expert's Industry Advisory Board. **Howell Huws** is a manager in the Knowledge Based-Systems Group at Coopers & Lybrand. They can be reached at Coopers & Lybrand, 1251 Avenue of the Americas, New York, NY 10020.