

## Book Reviews

### Information Coordination – The Management of Information Model, Systems, and Organizations

Richard Veryard

*Information Coordination – The Management of Information Model, Systems, and Organizations* is written by published by Prentice Hall 1994, ISBN 0-13-099243-7, 190 pp., \$40.00.

Richard Veryard writes in the opening paragraph: “This book is about coordination at three levels: the enterprise or organization, the computerized information systems that support the enterprise, and (in between) the information models of the enterprise. As we shall see, these models both belong to the enterprise and also represent the enterprise (or parts of it). Furthermore, when they have been used to plan and design computerized information systems, they represent the structure of these systems.” (p 1)

In this paragraph he sets the tone for his book – he suggests a very tight relationship between the enterprise and the information systems, and the models which tie them together.

In a recent study undertaken by CSC (see “Information Systems at Work in Business Reengineering”, InformationWeek, 10/23/95), they commented on three basic modes of operation to describe the interaction between Information Technology / Information Systems (IT/IS) groups and their business customers. What does this have to do with information models? CSC characterizes the first mode as the Provider/User. It essentially focuses on the IT/IS group delivering what the customer wants. The second mode they identified as the Partnership. In this setting, the IT/IS takes increased initiative to help the customers discover more clearly what they need, beyond what they simply want.

In the third mode, CSC suggests that the IT/IS group moves into a Pervasive mode of operations, becoming a fully integrated contributor to the business – able to not only react to business situations but to help discover the new challenges and opportunities. In highlighting this third mode of operation, CSC points out that it allows a business to quickly create flexible applications – because the business has an infrastructure featuring a rich and coherent set of business information models. Of course the organization must also have necessary expertise in exploiting technology to develop and deploy those applications. Each of these three facets of the Pervasive mode – modeling, deploying, and exploiting – stands in need of a Center of Excellence (CoE). But the differentiator lies in the modeling.

Veryard defines a model this way: “A model is a representation of the structure of information within an organization or part of an organization. It has a scope, whose exact boundaries are often difficult to establish. . . . It is important to be

aware of the perspective of a model: i.e., which stakeholders are included. . . . The intended purpose of a model influences its focus, and the appropriate style and detail.” (p 13-14)

So if an organization wants to move to the more sophisticated Pervasive mode of operation, the message of this book offers sound for guidance on the creation and maintenance of business information models. Just look at the chapter names – Introduction; Concepts; Planning; Planning Techniques; Development; Production; Model Management Techniques; Management. A CoE for the creation and maintenance of models will find these topics consuming much of its attention.

You can hardly pick up a trade journal without finding comments on the difficulties of producing software systems. Some of the authors dwell on causes – perhaps lack of certification of software engineers. Other authors cut immediately to solutions – perhaps a promising new technique. Still others will suggest thought-provoking analogies – perhaps comparing IT/IS to urban planning. Many point to the sheer complexity of the systems development environment – poorly understood or conflicting requirements, resource constraints, the increasing rate of change in the business environment, and on and on. Indeed, in a series of interviews with consulting firms, a task force in my organization heard one message several times about a particular emerging technique: When you consider object-oriented (OO) techniques, you will find that perhaps their major contribution lies in the reduction of complexity, more even than any contribution in re-use.

So does the migration to OO offer to solve the problems of model coordination? Certainly OO owes some part of that reduction in complexity to its focus on working with abstractions, with models. But Veryard insists that we will still face issues of coordination, but perhaps at a different level: “In the object-oriented approach, of course, the object is the main focus of attention. So we should expect to have object coordination rather than model coordination. Yet when there are thousands of objects in use across an enterprise, some grouping of objects into sets or clusters may be necessary to make this coordination manageable. (There has been surprisingly little attention given to this problem in the vast object-oriented literature, perhaps because most object-oriented development has been directed at stand-alone rather than portfolio systems.)” (p 129-130)

So Veryard suggests that we will continue to create models, using whatever techniques we can, in order to represent various perspectives for given purposes within some scope. (In an earlier book, Veryard spends his energies describing in more detail how to create the models, focusing apparently on the models featured in Information Engineering method.) But once we create these models we must coordinate them and manage them. Veryard reminds the reader that coordination falls into that group of attributes that we typically recognize only in their absence – we can surely tell when we don’t have it.

Thankfully, throughout the book, the author offers not doctrine but considerations. Veryard points out that not every



organization must try to achieve coordination in just the same way. That is, he does not offer the one right theory of coordination, but rather three theories: hierarchical, market, and network. Similarly, he offers three approaches to planning: top-down, organic, and exchange.

In those cases where he lists several alternatives, he notes some of the strengths and weaknesses of each, describes the utility of each, and suggests how to determine which may best fit a particular organization. For example, strongly centralized, hierarchically focused organizations may choose to employ a top-down approach – creating an all-inclusive enterprise model at some level of abstraction, then asking various projects to fill in the details over time. A more networked organization may choose to have the projects create the models, and depend on informal collaboration among staff member to discover commonalities. Then they can deliberately create interconnections after-the-fact. “The adaptive view is more pragmatic. Instead of trying to get the best possible consistency between the models, it contents itself with good enough consistency. This is what decision-theorists call satisficing, in other words making do with a less than perfect solution. . . . The advantage of this approach . . . is that it allows for some degree of organizational learning. New strategies and concepts can emerge and can be encouraged to emerge. However, there is still a feeling implicit in this view that the inconsistency is a regrettable imperfection.” (p 124-125)

Regardless of the techniques used to create the models or the scheme used to coordinate them, the organization clearly has an interest in achieving coherence among those models. The models exist in order to help achieve business goals. Veryard writes: “In most development programmes today, this vision of coherence is at best only partly achieved. . . . The best we can achieve allows the whole to grow in many steps, over many years. Its final form cannot be predicted, except perhaps in ambiguous generalities, partly because the form is sensitive to details that cannot (and should not) be worked out in advance, but partly (and more importantly) because the organization itself learns during the development: systems do not merely support the organization but transform it, or rather allow it to transform itself. . . . The success of a project should not be measured in the amount of activity of information technologists, nor by whether they are following this or that way of satisfying the business requirements, but by the satisfaction of the business requirements themselves.” (p 48-50)

In this context, then, an organization must strive to ensure that its models and its systems remain somehow connected – that is, the organization wants the capability to trace upwards from physically implemented computer code to the more abstract representations of the business. Veryard points out that this connection becomes even more important as the organization comes to greater dependence on computer systems. “Computerized information systems have the power to change the way the members of the organization think about their work. The more the organizations relies on computers, the more power the systems have. This is one of the reason

why the conceptual structure of the systems deserves serious attention from the management of the organization.” (p 77)

Veryard closes his book with these summary comments: “As we have seen in this book, the mission of centrally coordinated information systems, based on a single corporate data model, is not appropriate for all organizations. . . . The important thing is to fit the style of information coordination to the requirements of the enterprise, and not to a fixed ideology. Sensitivity and flexibility are crucial. Coordination may be complex and difficult to achieve, but it is not necessarily made easier by being made over-simple.” (p 189)

Reviewed by Michael Ayers, mbayers@mmm.com, (612) 733-5690, FAX (612) 737-7718, IT Education Svcs/3M Center, 224-2NE-02, PO Box 33224, St Paul, MN 55133-3224.

## Pattern Languages of Program Design

J.O. Coplien and D.C. Schmidt (eds)

*Pattern Languages of Program Design* is edited by J.O. Coplien and D.C. Schmidt, and published by Addison-Wesley publishing Company, 1995 (paperback) ISBN 0-201-60734-4, 562 pp., \$39.76.

The book is based on papers presented at PLoP, the First Annual Conference of Pattern Languages of Programs. Contents and form of the patterns presented are much more diverse than are the patterns presented in the Gamma et al book [1]. Where the Gamma et al book contains basic (concrete) design patterns only, the patterns in the Coplien/Schmidt book vary very much in size, abstraction level, application domain, and presentation form, i.e., the book is not a pattern catalogue. Fortunately the editors added an extensive index, which makes it easy to hit information fast.

The contents of the book is grouped into 8 parts. Each part is preceded by a very well-written and informative introduction relating the contents of the chapters in each part.

The book is not, as the title might suggest, a book on pattern languages describing solutions to design problems. It is much more than that. About half of the chapters describe single more or less concrete design patterns in the spirit of the Gamma et al book. The rest of the book deals with pattern languages, catalogs, or systems if you like. These chapters describe sets of patterns together with their interrelationships. I got the most value out of sections 4 and 5.

Section four (Process and Organization) does not contain common sense pattern languages, i.e., solutions for design problems. Section four contains cookbook approaches to model your organizational set-up and software development process. These patterns do not only contain the recipes, but also descriptions of rationales and deliverables. Chapter 15 describes the object-oriented requirements engineering process and can be highly recommended.

Section five (Design Patterns & Catalogs) summarizes, classi-