

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2003 Proceedings

Americas Conference on Information Systems
(AMCIS)

12-31-2003

Method Engineering: Reflections on the Past and Ways Forward

Duane Truex

Florida International University, dtruex@gsu.edu

David Avison

ESSEC Business School

Follow this and additional works at: <http://aisel.aisnet.org/amcis2003>

Recommended Citation

Truex, Duane and Avison, David, "Method Engineering: Reflections on the Past and Ways Forward" (2003). *AMCIS 2003 Proceedings*. 64.

<http://aisel.aisnet.org/amcis2003/64>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2003 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

METHOD ENGINEERING: REFLECTIONS ON THE PAST AND WAYS FORWARD

Duane Truex

Florida International University

David Avison

ESSEC Business School

Abstract

Method engineering (ME) is the process of designing, constructing and merging methods and techniques to support information systems development (ISD). It has taken many forms and is as old as ISD itself, and yet still the basic problems of controlling the systems development process remain. Method engineering is often associated with the hierarchical and bureaucratic approaches of the 1980s. But even in its most recent forms, as tools and capabilities embedded within enterprise resource planning (ERP) system infrastructures, problems still persist. This paper reflects on the genealogy of method engineering and look to how it must evolve to meet the needs of contemporary IS development.

Keywords: Method engineering, IS development, emergent systems development, information engineering, method engineering

Method engineering (ME) is the process of designing, constructing and merging methods and techniques to support information systems development (ISD). It has taken many forms and is as old as ISD itself, and yet still the basic problems of controlling the systems development process remain. Method engineering is often associated with the hierarchical and bureaucratic approaches of the 1980s. But in its most recent form, as enterprise resource planning (ERP) systems, problems still persist. Moreover, the business environment is changing such that information systems (IS) need now to cross organizational, cultural and political boundaries. Method engineering approaches need to adjust to meet the changing set of business requirements.

In this paper we reflect on the genealogy of method engineering and look to how it must evolve to meet the needs of contemporary IS development. As guides and as method engineering source books, we point the reader to Brinkkemper (Brinkkemper et al. 2000; Brinkkemper et al. 1996) and the April 2000 issue of Communications of the ACM (Kumar et al. 2000). The latter reflects in particular on the challenges provided by ERP systems. We will discuss these changes and point to potential directions for ME that are more likely to address the basic problems and needs of ISD. In so doing, we briefly review five phases in the ME movement, with the fifth being ERP systems, and suggest two possible future directions, one (described as type VIa ME) using ERP systems but making them fit the organization (rather than the other way round) and another (type VIb ME) independent of ERP. For the purposes of a debate it suggests that ERP configuration tools and facilities can be used in such a way as to be viewed as a kind of method engineering approach.

Early ME involved creating a standardized approach to 'engineer' systems development, usually based on process or data modeling, bringing order to work that had previously been largely trial and error and software oriented. In general, this contained techniques, phases and standards put together to form a coherent methodology to be used by systems analysts and programmers. As we see in Table 1, we refer to this as Type I method engineering.

Later, in what we refer to as Type II method engineering, ME was carried out to improve IS development methodologies so that they captured best practice by including techniques in other methodologies, such as dataflow diagrams and entity modeling, or take account of newly proposed ones, such as object-oriented methods. Many were extended to address more phases of the life cycle. The result was larger, perhaps bureaucratic, approaches. These widened the scope of methodologies to be more general-purpose and making them more commercially viable. Such methodologies were Information Engineering, Merise, SSADM and Yourdon Systems Method (Avison et al. 1995)

But this general-purpose, universal approach did not address the problems of complexity and of the skills required to develop IS. One response was the development of CASE tools. However, these have not necessarily increased productivity nor led to a better end product, though they have probably increased the expense of developing information systems. As a reaction to the complexity and inflexibility of methods, systems analysts frequently paid ‘lip service’ to a methodology, rarely following the phases exactly as described in the manual and sometimes omitting, adding or modifying phases. Some organizations stopped using methodologies altogether. One response from the methodologists was to define different paths within it so that a shorter path can be taken to speed up systems delivery, another to implement pilot projects and yet another to implement systems based on an application package. But these solutions also limit choice to well-defined project types, not different organizational situations.

Type III method engineering, the next development, links method fragments or components to form a ‘meta-methodology’. This aimed to increase the flexibility of approaches and the customization of a ‘one-off’ methodology for a particular application. One major difficulty is that the analyst is confronted with a ‘hodgepodge’ of techniques and tools, some of which may be inappropriately linked to others, thus most meta-edit approaches of this type provided a framework and control system to guide the analyst in choosing the appropriate method fragments. Some of these frameworks proposed sets of situational (or contingent) factors that guided the selection of method fragments (Slooten et al. 1996). However these frameworks tended towards a very technical view of the development of IS.

As **Type IV method engineering** recognizes, organizations are social constructs - they are about relationships between people - and artifacts do not determine how people behave. ISD is not just about choosing technique fragments. ISD was therefore broadened to include human and organization factors at least as much as technical ones. These approaches, such as Multiview (Avison et al. 1988), recognize that IS development is constrained by organizational context while providing the means to change that context. We cannot assume that technical improvements will lead to organizational improvement. Any IS development endeavor can suffer the unintended consequences of action. This suggests that IS approaches applied contingently require also that they be applied reflexively.

Although they answer some of the criticisms of earlier approaches, like its predecessors, there are weaknesses in Type IV method engineering approaches. The first weakness concerns problems relating to contingency itself. Techniques and tools come from different methodologies, which have different philosophies, and some argue that they should not be ‘mixed and matched’. The second concerns the burden on the systems analyst to choose the ‘appropriate’ techniques and tools for the situation. The third, in such an informal approach, concerns the control and application of standards in IS development.

Type V method engineering is the latest exemplar of the method engineering movement. Enterprise resource planning (ERP) systems, such as SAP, attempt to re-engineer whole organizations and, in effect, model an ideal organization (making the actual organization fit with this ideal). Like most of its precursors in the ME movement, it became fashionable but is now being critically scrutinized. (Truex III et al. 2001)

We propose that enterprise resource planning systems (Type V) represent both a type of ME approach and platform with design and configuration tools supporting that approach. ERP systems are built upon highly normalized and robust data models and industry-specific reference models purported to represent ‘best-of-class’ process and work flow models. As such they represent a composite of systems views and models ranging from data through process and work flow to object models and, ideally, seek to integrate all views such that data and process knowledge may be available to any of the application components wherever appropriate or needed.

ERP are a realization of ME goals in that they provide a consolidated and integrated tool-kit of method fragments and tools for modeling and building applications. The ERP vendors and the systems analysts enforce a framework by which the systems are brought on-line. ERP are not simply technologically centric because they incorporate business process reengineering and modeling tools. Nor do they simply assume given process models. For, depending on the management’s choice of implementation, they have very definite focus upon organizational considerations.

ERP systems are significant breaks with the past in two ways. First, for reasons well covered in the literature, most organizations find them so complex that they forego significant customization and effectively redesign the organization to fit the software system, frequently through business process re-engineering. This trend is wholly anathematic to past practice in which the goal of ISD has been to build systems that fit organizations, like custom-tailored gloves fit the hand. A second significant difference is that the driving force behind the acquisition of these systems tends to be *top corporate management* rather than IT management or even business unit management. Finally, as business process reengineering assumptions and values are a part of these approaches and systems, the focus has moved away from the integration of existing IS and the improvement or enhancement of

current systems, to a philosophy of wholesale replacement and abandonment of existing systems and ways of doing work. ERP are generally accompanied by organizational restructuring and significant reductions in headcount throughout the organizational hierarchy. Significantly the organizational reengineering follows the introduction of the ERP system rather than preceding its adoption. Thus it represents the instance of the software system calling the organizational tune as compared to a more traditional development setting in which the software is constructed to meld with the organizational demands.

There is a certain irony associated with ERP systems when they are seen as forms of method engineering. To some extent they hark back to the Type II ME approaches, which imply a universal view of organizational systems. The CASE tools associated with ME approaches enforced the sequence and the description of design upon the developer. So too do ERP systems, by virtue of the various models and components enforcing a kind of discipline upon the organization. A final irony is that if most organizations do not take advantage of the configurability of the system, then most organizations using ERP will look very much alike in terms of business processes and organization. Moreover, legacy systems tend to incorporate aspects of organizational distinctiveness and incorporated organizational memory. Many of these legacy systems have been abandoned with the advent of the new ERP system. However with the softening of ERP sales growth following the Y2K crisis methodologists and management alike began to consider how to overcome the inherent problems of this universal approach to how systems should be created.

ERP systems are a great deal more than a set of methods and tools; they are also regarded as infrastructures in that they incorporate a type of front-end, back-end and middle-tier architecture that runs on a host of different networks and hardware platforms. An initial appeal of these systems has been that they provided certain interoperability and sharing of enterprise data while allowing custom configuration of the applications set. This is a point that has proven to be problematic, and we return to this later. However, the sheer complexity and the cost to bring custom solution ERP up and running has proved so high that the pendulum has swung in the direction of standard, 'no change', implementation on rapid deployment product versions using special implementation teams. This brings the software up relatively quickly but does not address organizational issues such as restructuring and training. We are therefore at the stage of dissatisfaction with Type V method engineering. We now are faced with a strong demand for an alternative to ERP: we are in a post-ERP world.

It is our view that there are two possible directions for this post-ERP world. Both successor approaches must, of course, take into account the reality of the ERP infrastructure and ERP as a method engineered system. However, one approach adapts from within and the second either abandons or avoids the ERP approach altogether.

In **Type VIa method engineering**, the ERP is customized and configured continuously to fit in with the organization needs rather than accept the ERP notion of organizational structure and work process. Type VIa effectively attempts to make the ERP system an ideal system for the organization. The alternative, **Type VIb method engineering**, takes place outside the ERP infrastructure. It sees the ERP as but one of many potential organizational IT components to be knitted into an even larger network of components. As such, type VIb attempts to both incorporate and transcend all extant systems.

Method engineering Type VIa will retain the framework of system infrastructure with its data architecture, applications and network configuration, while seeking to either reflect or to establish a degree of organizational uniqueness via customized configuration of the ERP system itself. The focus in these systems moves away from the implementation of templates and 'best-of-breed' systems towards using the facilities that ERP vendors provide to customize components and applications. A variation on this will be the purchase of third-party tools and the creative adaptation of applications coming from outside the industrial reference models commonly associated with the particular ERP systems. In the best of worlds this approach will allow organizations to express a natural tendency to emerge from under any fixed information system. In other words, the organizational is a living changing entity which transcends and confounds any fixed description of that system, such as software. In an earlier CACM article (Truex et al. 1999) illustrates how traditional systems development does not keep up with organizational emergence and suggests all enterprise information systems to be continuous prototypes and as never-finished artifacts.

Once organizations have a wholly integrated application portfolio built on common platforms, then more-or-less continuous subtle configuration may allow for systems which for the first time may track rather than inhibit organizational emergence. It may therefore be possible to build robust systems which by continuous adaptation will not be broken by this organizational change. Where in each of the previous examples the primary orientation has been to either technologists or managers, in this scenario we may have the opportunity for a kind of 'joint optimization' between the ERP system and the organization, albeit within the constructs of a given ERP infrastructure. In this instance managers may adjust business models in response to changing environmental, technological and competitive circumstances with a high confidence that systems can keep up and will not inhibit such constant adaptation. The ERP method engineer and developer may therefore adapt to a world where component integration and configuration management replace systems development and redevelopment. Hence one may actually model and maintain the ideal system for the particular organization.

<i>Type</i>	<i>Focus</i>	<i>Intent</i>	<i>Added Value</i>	<i>Orientation</i>	<i>Examples</i>	<i>Metaphors</i>
Type I	Uniview	Standardize on best practice in systems development	Bring order to chaos, structured, methodological approach	Technical Technologist	STRADIS (process view) D2S2 (data view)	Modeling data <u>or</u> processes
Type II	Making approach more ecumenical (filling in the gaps)	Connect univiews to engineer an overall approach	More generalizable, universal More complete	Technical Technologist	Information Engineering (Accelerator CASE) SSADM Merise	Modeling data <u>and</u> processes
Type III	Identifying and linking method fragments	Add technical richness by adding modules or method components	Broaden scope further Ability to inter-connect methods into whole 'super (meta)-method'	Technical Technologist	Meta-Edit (CASE)	Modeling various systems components
Type IV	Contingent frameworks	Provide guidance to use techniques Bring in organizational and social richness	Provides guidance about how and why we use method components Broaden scope further	Social and organizational as well as technical Middle managers and users	Multiview	Modeling the proper match of components
Type V	Enterprise resource planning	BPR – Re-engineer whole organizations (to fit the system?)	Provides more structure to organizational frameworks Adds BPR and workflow	Organizational and technical Top managers	SAP, BAAN, Peoplesoft	Modeling the ideal organization
Type VIa	Post ERP (within ERP infrastructure)	Customize & configure organization & system needs continuously	Technical system fits in with organization (not the other way round) Emergent contingency Regain competitive advantage by customizing Gain of control	Organizational and technical Technologist and managers	Highly customized (after ASAP)	Modeling the ideal system for the organisation
Type VIb	Post ERP (outside ERP infrastructure)	Bridge multiple types of data and knowledge repositories	Extend domain beyond data to multiple forms of knowledge, more robust and long term	True organizational All actors	?????	Modeling the whole organization (knowledge, history, memory)

Figure 1. The Development of Method Engineering

There are of course limits to this approach. Intra-organizational sharing of systems output and data, for instance in the case of EDI or business-to-business electronic commerce, requires a high degree of standardization. By customizing the ERP's process and data models a company is abandoning the safe harbor of the ERP standard in favor of a model more like the environment it left

behind. In a way, this treats the ERP as if were an application ‘superset’ rather than an integrated whole. In any event, to take advantage of the ERP standard it now becomes essential to translate back from the customized system to the ‘standard’ to adjust for customization. But even the notion that the ERP represents a standard that is widely shared and accessible is illusory. What happens when the desired trading partner uses a different ERP ‘standard’? Thus the notion of a standard is itself called into question such that we need to be more aware of actual limits.

For instance, even national or industry-wide standards, such as the UN EDIFAC or ANSI X12, cannot serve to define the particulars of any interaction between a pair of trading partners. Rather the standard may only map out the types of things that might be included in such an exchange (Brousseau 1994). The particulars of an essentially goal-driven need to share data between trading partners and the partners’ particular work practices, as well as data they are trying to share, and not the general standard, will determine the precise final form of the exchange. Damsgaard and Truex (Damsgaard et al. 2000) provide an illustration of how EDI is a kind of organizational discourse depending on a particular form of work language with its own emergent grammatical forms).

Because ERP systems allow high standardization and, theoretically at least, allow high customization, it would appear that they overcome the problems of earlier ME approaches. Yet the very strengths of ERP systems also represent a type of ‘Achilles heel’ for them. They are based upon highly centralized models of organizations, of management and of system architecture, and ultimately of IS development. Moreover their highly normalized and centrally administered databases are not consistent with highly-distributed data, processes and presentation which are the *lingua franca* of Internet-based protocol systems typical of the evolving e-commerce world. ERP vendors are of course working to adapt to a less-centralized world. But these products were born of another highly centralized age and the jury is out as to whether they will be able to make the transition. SAP’s version 4.6, for example, attempted to solve this problem, but early market response suggests that it has not done so very successfully. In any event, the ‘ubiquitous, any-time, any place view of e-commerce’ suggests that what is needed is an ME model capable of constructing the simultaneously centralized and decentralized, top-down and bottom-up, stable and relatively permanent and yet nimble and discardable information systems. This is a very difficult goal to satisfy, and an ERP environment, even a ‘new’ ERP system, is not convincing in this role.

A potential answer to this problem is a **Type VIb ME** approach. Truex et al. (2000) illustrates our metaview of current business systems applications as they relate to the repositories upon which they depend or populate. This is referred to as an organizational memory information system as we see knowledge management as the core service of information systems, enabling capabilities to be built on the resources of the organization. In other words, best practices can be shared within the organization whilst making its expertise more valuable and difficult to imitate externally. We also see data warehouses as being much more valuable than databases, subsuming their capabilities amongst others. The data warehousing and data mining tools need to make effective use of the potentially valuable warehouse. Method engineering principles aim to ensure that all the inter-related ‘pieces of a jigsaw’ found in figure 2 integrate into a meta-system which fits into the particular organization.

A more ‘web enabled’ infrastructure view of Type VIb ME is to be found in figure 3. This subsumes the traditional mix of application data as ‘legacy systems’ thereby assuring their availability and partnership in the organizational portfolio. But this view removes them from the central focus in favor of an ‘Internet glue’ for various system elements. In this second view we see that the Intranet is added as a kind of bridge or linking technology which on the one hand allows for independent ‘quick hit’ development efforts and also ties into and captures the power of more stable organizational applications and data sources.

We see this Type VIb of method engineering as integrating ideals and systems and organizational elements in a way that was not possible previously. Although we see Type VIa ME as the most likely next step beyond Type V ME because it develops on the previous ERP technology, we see type VIb approach as having greater scope to address the basic problems of ME, and IS development generally, which were discussed earlier in the paper. In both Type VI approaches the whole is indeed greater than the sum of the parts as the subsystems adapt to the organization’s requirements, but Type VIb provides greater flexibility and other advantages as the organization is released from the ERP straitjacket.

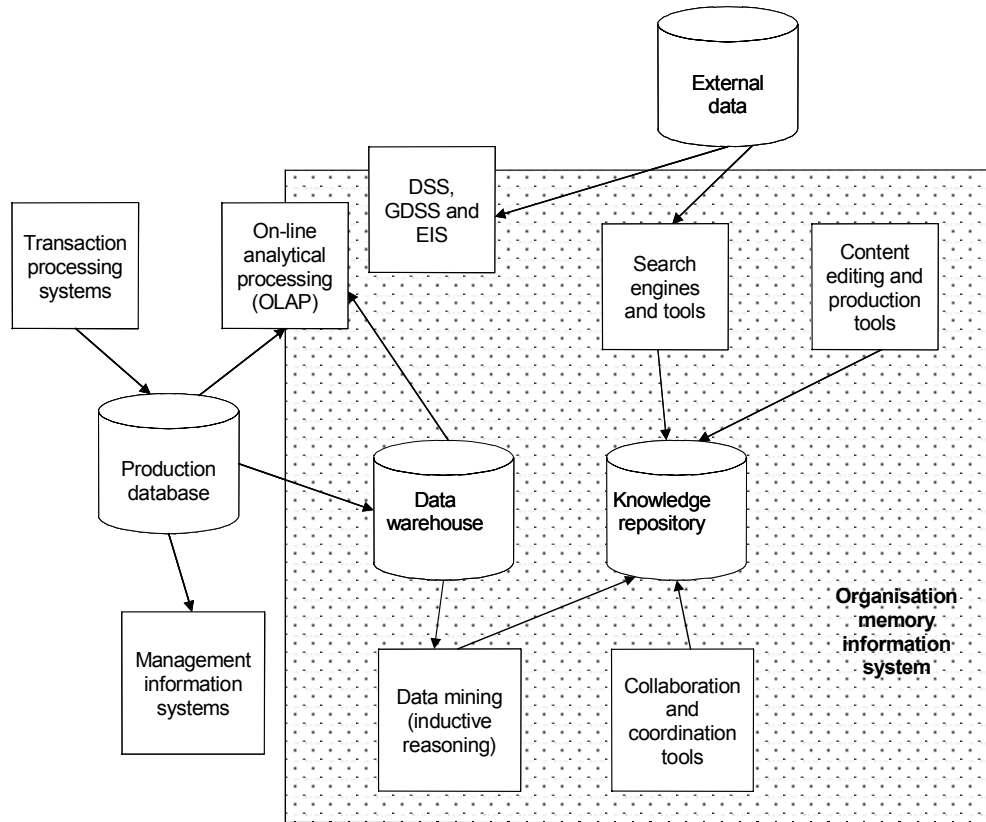


Figure 2. An Integrated Business Information System
(adapted from Raven and Truex 1999)

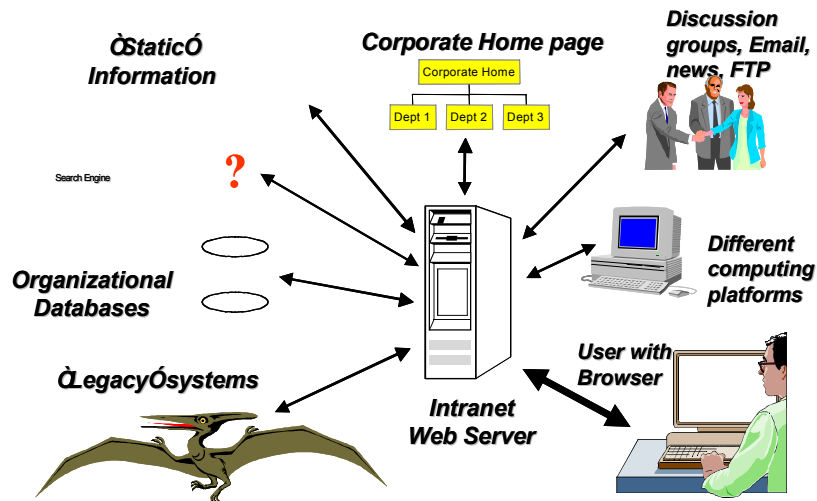


Figure 3. The Intranet Within the Organization (Damsgaard and Truex, 1999)

References

- Avison, D.E., and Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools* Blackwell Scientific Publications, London, 1988, p. 323.
- Avison, D.E., and Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools*, (Second edition. ed.) McGraw-Hill, Maidenhead, 1995.
- Brinkkemper, S., Lindencrona, E., and Solvberg, A. (eds.) *Information Systems Engineering: State of the Art and Research Themes*. Springer-Verlag, London., 2000.
- Brinkkemper, S., Lyytinen, K., and Welke, R.J. (eds.) *Method Engineering: Principle of Method Construction and Tool Support*. Chapman & Hall, Atlanta, USA, 1996.
- Brousseau, E. "EDI and inter-firm relationships: toward a standardization of coordination processes?," *Information Economics and Policy* (6) 1994, pp 319-347.
- Damsgaard, J., and Truex, D.P. "Binary Trading Relations and the Limits of EDI Standards: The Procrustean Bed of Standards," *European Journal of Information Systems* (9 #3:September) 2000, pp 1-16.
- Kumar, K., and Hillegersberg, J.v. "Introduction to a Special Issue on Enterprise Resource Planning Experiences and Evolution," in: *Communications of the ACM*, 2000, pp. 22-63.
- Slouten, K.v., and Hodes, B. "Characterizing IS Development Projects," in: *Method Engineering: Principle of Method Construction and Tool Support*, R. Welke (ed.), Chapman & Hall, London, 1996, pp. 29-44.
- Truex, D., and Raven, A. "A Data-centric View of Organizational Information and Knowledge Systems," 2000.
- Truex, D.P., Baskerville, R., and Klein, H.K. "Growing Systems in an Emergent Organization," *Communications of the ACM* (42 no. 8:August) 1999, pp 117-123.
- Truex III, D.P., and Howcroft, D. "What is a Critical Social Theoretic Approach Towards Research on ERP systems?: An introduction to the special issue on a Critical Social Theoretic view of ERP," *The DataBase for Advances in Information Technology* (32:4:Fall 2001), December 2001 2001, pp 1-9.