AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

From the Perspective to the Analytic: A New Direction for Information Systems Engineering Methodology

Galal Galal
University College London

Ray Paul
Brunel University

From the Prescriptive to the Analytic: A New Direction for Information Systems Engineering Methodology

Galal H Galal
Software Systems Engineering Group
Department of Computer Science
University College London

Ray J. Paul
Department of Information Systems and Computing
Brunel University

Abstract

Information systems engineering (ISE) methodology has been so far predominated by an overly normative/ prescriptive view of what methodology should be. This has been to the detriment of both the contextual fit and dynamic evolvement of an information system. This paper argues for the analytic emphasis to supplant the prescriptive one, so as to enable the concept of continuous design needed for the dynamic evolvement of information systems.

Introduction

In this paper, we progress our issue-oriented discussion of methodology (Johnson & Galal, 1993), (Galal, 1997) a step further. This time, we are concerned with the epistemological emphasis of information systems engineering methodology. Hillier (1996), in discussing the role of theory in architectural design argues that architectural theory contributes to design through two distinct aspects: the normative aspect tells the designer where to look for candidate designs in the creative phase; and the analytic to help him or her predict how a particular design will work and be experienced. We discuss the salient directions in information systems engineering methodology from the normative/ analytic angle, commenting on the direction in which we believe emphasis should be shifted. However, we prefer to extend the "normative" attribute by the "prescriptive" attributes, on the assumption that prescription follows a specific, normative view of where designs should be sought, and how the information systems engineering process should be conducted. This is an issue that lies beyond the fundamental ontological assumptions of information systems methodology (see Hirschheim & Klein, 1989), but is also based on it. We explain why an analytic emphasis is more suited to the highly dynamic contexts which most substantial information systems find they need to satisfy.

The Methodology Debate: ALittle History

The information systems engineering methodology issue has probably occupied the minds of both researchers and practitioners of information systems more than any other single topic in the area. Methodology proposals ranged from the highly prescriptive type (Downs et al., 1992; Gane & Sarson, 1979; Jackson, 1983, to name but a tiny proportion), to the more open-ended, tool-box like, but still essentially prescriptive, type (e.g. Avison & Wood-Harper, 1990; Checkland & Scholes, 1990). We might say that this prescriptive orientation is based upon a normative view of how systems engineering should be conducted. Prescriptive methodologies tend to rigidly specify particular system modelling tools and techniques for representing and validating systems requirements and architectures, such as Data Flow Diagrams, Entity-Relationship Diagrams etc. Thus, the evaluative/ analytical aspects of these methodologies focused on the micro-scale, namely; constructional aspects of systems such as data structures and algorithms. More open-ended methodologies tend to suggest a collection of modelling tools and techniques, along with particular configurations -or sequences- of these, but also suggesting that the systems engineer adds or subtracts from such tools as he or she sees fit, as long as the fundamental analysis and/or design perspectives the methodologies advocate are preserved.

Non-sequential Process Models

However, engineering information systems has its processual, as well as content-modelling, side. The processual side is what may be termed the *engineering process model*. Independently from content models, engineering process models that depart from assuming apriori full specification have been proposed. Some of the best known are the Prototyping model (Brooks, 1982; Carey, 1990), the Evolutionary model (Gilb, 1988), and the Spiral model (Boehm, 1988). All these models rely, to varying degrees, on a limited amount of system engineering, a prototype or a small increment, deploying and evaluating it, and using the result of the evaluation to guide further system engineering efforts, such as requirements, architecture, interface design, performance attributes, and so on. In the case of the Spiral model, the process is a configurable entity in its own right. We refer to this category of process models as *non-sequential process models*, since they adopt cyclic or iterative models of the systems engineering process. It is in these non-sequential process models that attention to the macro-analytic aspects relating to system-wide attributes, its architecture and fit, of methodology started to emerge. Also, these process models give the systems engineer complete freedom in choosing the particular content modelling tools and techniques. As such, these process models highlighted the distinction of process from content-modelling issues. Any of the process models that we refer to above can be followed independently of whether a Data Flow or an Object model is best for modelling the system in question. This is how the methodology and process issues can, and should be, de-coupled at least initially and conceptually.[1]

In any case, a key factor in the success of non-sequential process models is the quality and comprehensiveness with which the evaluation/ analysis activity is carried out. The evaluation of a system increment or a prototype needs to encompass aspects of the system's dynamic context to allow early adjustment of the various systems engineering decisions.

From the Static to the Dynamic

The problem with the above approaches to information systems engineering -whether concerned with process or content- is that when they allow for the gradual evolvement of requirements, they assume that there is a point in time when all stakeholders' views will converge, producing a stable and final view of the system's requirements. This is what Paul (1993) terms the *fixed point theorem*. Even in evolutionary/ incremental models, there is usually an end point to the engineering of the system, even after frequent adjustments to the systems' objectives. These fixed-point oriented view neglect the fact that complex information systems exist in highly dynamic and fast-changing contexts. Organisations are dynamic entities, which necessitated that any technological systems hosted by them will be judged against highly dynamic realities (Kanellis & Paul, 1996). Unfortunately, the prescriptive/ normative orientation that predominate most current methodology and process model proposals do not deal adequately with these realities. What is needed are analysis/ evaluation oriented methods that can be continuously applied to the macro (or architectural) scale of information systems during the engineering process itself, rather than waiting until a system is delivered. The analytical element needs to semantically test the evolving models achieved through applying the methodology. Our view is consonant with that of Monarchi & Puhr (1992) who argue for a stronger *semantic*, rather than technical evaluation for models developed by object oriented methods. The semantic evaluation aims to analyse the technical models against the domain of interest. Likewise, from a "whole system" point of view, we need methods for evaluating system models by analysing their technical merit *during* development and not merely after. For the whole system, this goes beyond merely validating or verifying putative micro-scale models, showing for example the data-structure view of the informational artefact. We need to develop and analyse a vision of how the whole informational artefact may fit its context, and satisfy its intended purpose. This should be done with full consideration of the dynamicity of the contexts of evolving information systems. In essence, we are calling for the prescriptive/ normative element of methodologies to be tempered by a more analytic element applied to the larger scale view of information systems.

Also needed for the dynamic evaluation to be truly useful is methodical or investigative discipline, achieved through *rigorously grounding* the evaluative/ analytic frameworks in the investigations and analyses of the information systems being engineered.

Which Analysis and Evaluation?

The challenge that faces the information systems engineering community lies in developing analytic frameworks, supported by explicit theoretical perspectives, that serve the analytic intent of the systems engineering practice. All while focusing on macro, architectural aspects of the systems being developed: semantic quality and fit; overall, emergent system effects; and dynamic potential. The techniques required should also pay particular attention to the qualitative data that characterise the dynamic contexts of the systems being engineered (Galal, 1997), perhaps using some scenario or vision-oriented representations. However, the analytic mechanisms used need to be compatible with the technical models being developed of the system, for example by utilising the same concepts used the requirements' model. This facilitates using the results of the evaluation in revising the technical models. Otherwise, the results of the analytic evaluation may be thrown away rather than being integrated into the technical engineering process due to difficulties in integrating the two. For example, a good analytic practice may lead to the design of new functions, modification of required performance attributes or, perhaps more importantly, to the re-design of appropriate parts of the target environments that is to host the engineered system. We are talking about total systems engineering in the full sense of the word.

[1] This should be a minimal response to colleagues and amply experienced practitioners who always remind us that methodologies don’t work because you can’t decide on all of the system’s requirements and architectures at the start: this is not a necessary consequence of adopting a methodology.

A Question of Architecture

The technical requirement for rendering the view we put forward in this paper operational is that the system architecture needs to be as open, and as extensible, as possible. In Gilb's terms, there must a global, open architecture that is minimally committal, to allow changes in strategic directions without nullifying earlier engineering deliverables. From a software engineering angle, similar views have been expressed for layering the software architecture to localise changes as far as possible, making modifications easier, cheaper and safer to effect (e.g. Coad & Yourdon, 1991; Jackson, 1983). The same views need to be operationalised from a system-wide perspective, rather than mere software modules as these are only constructional elements as viewed from the whole system perspective. So, from a technical point of view, the architectural design of the system needs to reflect a *continuous engineering* philosophy, whereby at no point during the useful life of the system, the engineering process is deemed to have been concluded. This philosophy requires changes in the mind-set and attitudes of both the systems engineers and clients, which needs its own change-management efforts!

A Way Forward

Our particular approach to address the concerns that we outline above taps the potential of the Grounded Theory method (Glaser & Strauss, 1967), as an interpretive framework for developing dynamic views of putative information systems, against which system increments may be evaluated (Galal, 1996). We thus serve the analytic view that we advocate, as well as allow the results of the evaluation to be directly fed into technical systems engineering decisions, leaving the narrow evaluation of micro-scale models to the more constructional aspects of the system, and focusing on the more architectural, system-wide ones. We should also like to mention that our particular framework (GSEM, the Grounded Systems Engineering Methodology) relies predominantly on qualitative data to construct theories that express predictions about the system, while being at the same time interpretive from the paradigmatic point of view, since situation-bound interpretations can be easily incorporated into the type of theories that can be developed (Galal, 1996). We hope that our framework will be the subject of future writings of relevance to the practice-oriented, as well as the philosophy-oriented segments of the information systems engineering community.

Conclusions

Much of the methodological debate in information systems has been dominated by the prescriptive/ normative views of information systems engineering. We argued in this paper that balance needs to be restored by enhancing the analytic/ evaluative aspects, while focusing on the whole information system, rather than narrow technical aspects for it. Our views have been reflected earlier in both the software and information systems engineering communities in what we term non-sequential process models, that fundamentally rely on an iterative cycle of engineering, evaluation and feedback. The important thing is that the evaluation and feedback cycle needs to take place during development while being grounded in the dynamic contexts of information systems from an architectural, rather a narrow, constructional point of view.

References

References are available upon request from the authors (Galal@acm.org; Ray.Paul@Brunel.ac.uk).