Essential Principles for Workflow Modelling Effectiveness*

A.P. Barros[1], A.H.M. ter Hofstede[1], H.A. Proper[2]

[1]Department of Computer Science
The University of Queensland
Brisbane, QLD 4072
Australia

[2]Faculty of Information Technology
Queensland University of Technology
GPO Box 2434, Brisbane QLD 4001
Australia
E-mail: E.Proper@acm.org

PUBLISHED AS:

A.P. Barros, A.H.M. ter Hofstede, and H.A. Proper. Essential Principles for Workflow Modelling Effectiveness. In G.G. Gable and R.A.G. Webber, editors, *Proceedings of the Third Pacific Asia Conference on Information Systems (PACIS'97)*, pages 137–147, Brisbane, Australia, April 1997.

Abstract

While the specification languages of workflow management systems focus on process execution semantics, the successful development of workflows relies on a fuller conceptualisation of business processing, including process semantics. Traditionally, the success of conceptual modelling *techniques* has depended largely on the adequacy of certain requirements: conceptualisation (following the *Conceptualisation Principle*), expressive power (following the *One Hundred Principle*), comprehensibility and formal foundation. An equally important requirement, particularly with the increased conceptualisation of business aspects, is *business suitability*. In this paper, the focus is on the suitability of workflow modelling for a commonly encountered class of (operational) business processing, e.g. those of insurance claims, bank loans and land conveyancing. Based on a previously conducted assessment of a number of integrated techniques, the results of which are summarised in this paper, five business suitability principles are proposed: organisational embedding, scenario validation, service information hiding, cognitive sufficiency and execution resilience. As a result, a further insight into workflow specifications and workflow deployment in open distributed architectures is claimed.

1 Introduction

The workflow concept, proliferated through the recently emergent computer supported cooperative work (CSCW) systems and workflow systems (see surveys in [FYW94, WW93, Rod91] and [GHS95] respectively), advances information systems (IS) implementation models by incorporating aspects of

*Part of this work has been supported by CITEC, a business unit of the Queensland Government's Department of Public Works and Housing (formerly the Administrative Services Department).

collaboration and coordination in business processes. Under traditional implementation models, applications are partitioned into discrete units of functionality, with (typically) operational procedures used to describe how human and computerised actions of business processes combine to deliver business services. Through an enhanced endowment of business processing, workflows permit a greater organisational fit of ISs. Moreover workflows are specified at a level above traditional applications, enabling program binding and access to a loosely-coupled set of databases and files. Therefore, newer applications may be developed out of existing applications to reflect reengineered business processes.

Crucial to the specification of any IS implementation is the *conceptual* level. This, of course, orients the analysis of a given domain towards its essence (deep-structure) rather than to aspects of implementation (physical-structure) or representation (surface-structure). For workflows, the standardisation of concepts is progressing through the Workflow Management Coalition[1]. While the set of terms and references defined so far characterise sufficiently the notion of workflow, e.g. event, process (including pre-conditions, post-conditions and state transitions) and organisational (or actor) role, much of the focus is geared towards workflow management systems and their specification languages. The emphasis is on part of business processing, namely process execution semantics: sequence, repetition, choice, parallelism and synchronisation. A sound conceptualisation requires not only this but also that process semantics, e.g. the messaging, database updates and retrievals involved, to be explicitly captured.

In general, for the conceptual level, a plethora of techniques have emerged under different paradigms. When integrated into well-formed methods, integrated IS specifications should result. Several modelling paradigms may be discerned for workflow modelling: process-centric, e.g. [DB91]; state-centric, e.g. [DP95]; and actor-centric, e.g. [Die94] (based on the speech-act theory synthesis of [FL80]). Moreover the use of business (or enterprise) models, e.g. as deployed in requirements engineering methods [BB95, LK95, AMP94], in design methods [Ram94] and in CAiSE tools e.g. AD/CYCLE [MMNR90], provide an *organisational embedding* whereby a workflow model's components may be backtracked to its real-world counterparts.

For a sound conceptualisation, a technique should fulfill a set of often-cited (albeit various in terminology) requirements. If not properly catered for, the development of workflow specifications may be problematic. First and most naturally, the *Conceptualisation Principle* [ISO87] positions a technique on essence. In this regard, business models of workflow specification languages can prescribe organisational infrastructure, thereby compromising conceptualisation, while an execution-only focus can deflect it altogether. Conversely, following the *One Hundred Percent Principle*, techniques should provide a sufficient *expressive power* so that a full conceptualisation is in fact possible. If the expressiveness of a workflow specification language exceeds that of a technique to the point where only a partial conceptualisation results, the remainder has to be addressed at the implementation level. This situation, known in software engineering jargon as the *waterfall*, leads to the relocation of certain parts of analysis to the implementation level. Conceptual models also need to be communicated and validated with a diverse set of stakeholders and so mechanisms are required for an effective *comprehensibility*. An effective graphical presentation together with abstraction/decomposition mechanisms facilitates this. At the same time, a *formal foundation* is required to prevent interpretation ambiguities and to enable formal reasoning. That a recent survey on workflow technology [GHS95] cited the problem that "workflow models and process methodologies do not explicitly support the specification of what it means for a workflow to be correct" is symptomatic of a lack of formal semantics. Together with a formal syntax, this constitutes formal foundation.

[1] Refer to http://www.aiai.ed.ac.uk/WfMC/index.html for more details.

A final requirement which is not often salient in techniques, is their *suitability* for intended problem domains. This follows from the fact that a "silver bullet" for all types of domains is considered unrealistic (see e.g. [BS87, Bro87, ML83]). Suitability implies a close connection between modelling concepts and features, and those required by the domain. Of course, the more suitable a technique, the more effective the model generated through it. In absence of a universal organisational theory, the business suitability of techniques can only be understood through incremental *constructivist* perceptions - empirical and theoretical [FHL$^+$98]. As such, any world-view belonging to a technique, can only, at best, embody *inter-subjective* views of "reality". Clearly, the more satisfactory the constructivism, the more suitable the technique.

The focus of this paper is on the *business suitability* of workflow modelling. Of course, to speak of a general business suitability is vague since there are many types of organisations and many types of business processing [DO85]. Rather, particular attention is drawn to a specific type of (operational) business processing which exhibits precise execution paths in accordance with strict operation procedure, e.g. the processing of insurance claims, bank loans and land conveyancing. In [BHPC96], an assessment of a number of integrated techniques was conducted using the aforementioned requirements. In this paper, the ramifications for business suitability - important principles elicited from the assessment - are presented. These are: the organisational embedding of conceptual models; their validation (in general, their cognition) through scenarios - as proposed, through *business transactions*; the insulation of business service requests from the resultant business processing, and its motivation for service modelling within workflow specifications; the incorporation of sometimes neglected aspects, e.g. combining structural and behavioural aspects of processes, interactive aspects, in particular, human to computer interaction (HCI), and temporal aspects, to enhance a workflow's cognition; and the explicit treatment of operational error handling for a workflow's execution resilience.

The paper is organised as follows. In section 2, a summary of the assessment is presented in terms of well-known paradigms - structured process modelling, object-oriented modelling, behavioural process modelling and business-oriented modelling. In section 3, the problems and business suitability principles are defined. In section 4, the paper is concluded and the related research strategy is presented.

2 An assessment of integrated conceptual modelling techniques

In this section, a summary of the assessment of integrated techniques classified into - structured process modelling, object-oriented modelling, behavioural-process modelling and business-oriented modelling - is presented.

Structured process modelling

Structured process modelling techniques, for example Structured Analysis [GS86, You89, DeM78] and ISAC [LGN81], provide a top-down process analysis and software design mapping. Processes form the focal units of cognition, propagating data flows (messages) and updating data stores (persistent storages of data). As such a *process-centric* integration with data modelling techniques is apparent, i.e. the process model is more pivotal to the understanding of the collective conceptual model. Moreover, the integration of the techniques is *loosely-coupled*, i.e. each partial model has a separate definition, and some concepts may be integrated for basic consistency.

The simple and general set of concepts together with the feature of decomposition are applicable for different types of domains. At higher levels of analysis, models such as Data Flow Diagrams (DFDs), are easy to comprehend and are easily applicable for business domains. A precise definition of what aspects of organisational processing structure should be modelled and how, is not available. That only basic aspects of business processes can be modelled is indicative of a coarse business suitability. At the lowest level where the focus is on detailed design, pseudo-code is used for process specifications. The loosely-coupled integration structure, however, restricts expressive power. In general, the procedural style of conceptualisation forces a more imperative and less declarative approach to specifications (than in behavioural process modelling techniques). Also process execution dependencies are not considered other than the process sequencing implicit in module design mapping. A major deficiency is the lack of a formal foundation, allowing ambiguities and inaccuracies in process models.

Object-oriented modelling

OOA/D techniques, e.g. [SM88, Boo91, CY90, RBP$^+$91, JCJO92], *tightly-couple* object structure and behaviour into an object model, citing as a motivation the "natural" occurrence of objects in most types of domains. A *state-centric* integration is advocated principally, where the object model is most pivotal. Object lifecycles (typically finite state machines), defined for object types, allow a state-based specification of object behaviour. An event-action-condition language is typically used for such specifications. Through this, a large class of business rules may be captured declaratively. At the same time, imperative constructs (e.g. choice and iteration) are still required to augment conceptual level specification languages. In domains where object interaction is high, as tends to be the case in business domains, higher level contexts through process models (e.g. DFDs) or user scenarios (cases) are required. These extensions further necessitate an overall formal foundation, however little evidence of formal semantics is apparent in OOA/D techniques. The resolution of difficult design decisions such as complex object aggregation and method classification through higher level models is therefore intuitive. Again, generality inhibits business suitability.

Behavioural process modelling

The behavioural aspect of process models fundamentally concerns execution dependencies between processes, i.e. control flow. This includes the sequence, repetition, parallelism and synchronisation of processes. Despite generality, these features allow greater semantics of business processes to be captured. Traditionally, given their precise and graphically communicable operational semantics, Petri nets have been used to develop behavioural process modelling techniques, e.g. [SK86, RD82]. As such, model validation through execution is also possible. The inclusion of states in process models - note, still process-centric - provides a declarative approach to process specifications, and therefore to the specification of a further class of business rules. However, the impact on the graphical representation is problematic, with process models becoming cluttered for even basic specifications. Strategies to alleviate this include the use of abstraction/decomposition, e.g. [Kun93] uses DFDs, and the graphical restriction to process triggers only, e.g. Task Structures [DB91]. For the latter, detailed specifications rely on a highly expressive conceptual specification languages and a well-defined formal semantics. For example, in [HN93], the semantics of Task Structures are defined in Process Algebra, and a conceptual specification languages is defined in [HPW93].

Business-oriented modelling

Business-oriented techniques specialise conceptual modelling so that a precisely defined business suitability is achieved. For this, an alignment of concepts with organisational processing structure and a provision of collaborative processing are required. Some techniques cater for a divergent set of business world-views through a basic, but not necessarily complete set of business constructs, e.g. [DP95]. Other techniques, e.g. [Ram94], prescribe a particular world-view - a business (or enterprise) model. Business models abstract from the details of business plans of organisations. Business plans include the operational and strategic structure through qualitative descriptions of mission, goals, objectives, critical success factors, market sectors, competitive and quality management strategies. These qualify the business services provided and organisational processing structure designed to carry out the business services. Business models generally focus on basic aspects of organisational processing structure: goal-rooted organisational units, business services, activities, tasks, actors and actor roles (for processes) and resources (information or material).

Unless otherwise addressed, the quality of conceptualisation, expressive power, comprehensibility and formal foundation are inherited from the "underlying" classical techniques. In integrated specification environments, IS models are refined from business models and so the formal semantics of the IS model should relate closely to that of the business model. Also, the structural dependency of IS models on business models can occur, therefore violating the Conceptualisation principle. Communication-based techniques, e.g. [Die94], focus on organisational actor-based communication (a speech-act synthesis of actor communication). This provides a more essential insight into business process redesign, although its additional value for detailed IS design is not well-understood as yet.

3 Business suitability principles

Following the general and comparative insight obtained for the various approaches and (in)adequacies of a representative sample of integrated techniques, the problems of business suitability are now identified for workflow modelling. As a result, five business suitability principles are elicited.

3.1 Organisational embedding

The first problem relates to the relationship between business models and conceptual models. On the one hand, the adaption of classical techniques with some business-oriented constructs leaves room for an arbitrary relationship. On the other hand, the incorporation of business models might well lead to the situation where organisation processing structure prescribes the essential structure of the conceptual model; e.g. when an IS model is decomposed hierarchically from a business model. This violates the Conceptualisation Principle because the over-riding physical-structure compromises the generalisation, classification and specialisation of constructs often associated with deep-structure. In particular, as illustrated in Figure 1, it can preclude the design of an IS process from a network of business processes without first composing an "artificial" business process.

This restriction is evident in business models of integrated modelling techniques, e.g. [Ram94]. Such a composition at the IS level and not the business level relates to the comprehensibility of a process model, or reflects IS design requirements (of a multi-organisational domain say). To avoid these problems, the following principle is proposed:

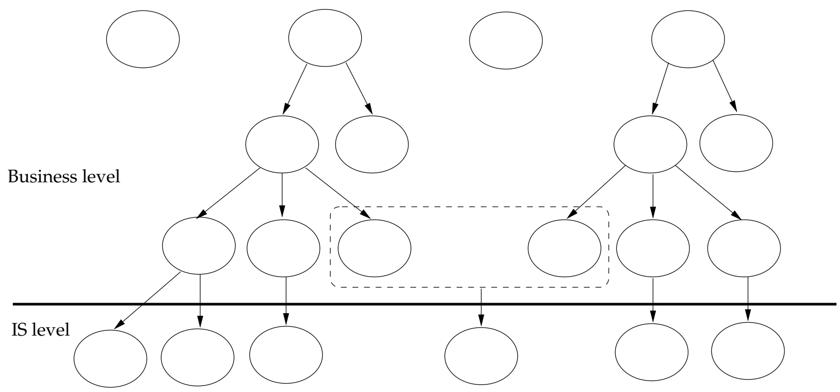

Figure 1: A violation of conceptualisation in business to IS model refinement

Principle – *Organisational Embedding*

A technique should embed all concepts in a conceptual model, directly or indirectly, but without redundancy, into organisational elements. □

Note, the principle does not prescribe the use of business models. Rather, it states that any modelling concepts be backtracked to organisational elements (whatever form of their definition). It can be seen that dependency of IS models on business models is replaced by an inter-dependency. That is, IS and business models are permitted their own modelling autonomies, and a consistency between the two is required. The consistency does not preclude mechanisms such as networked decompositions. Therefore, business task compositions are permitted for IS process design.

3.2 Scenario validation

The second problem stems from the observation that all techniques deal, at best, only partially with the validation of models. Validation is concerned with ensuring that a conceptual model is indeed a model of a business domain. Beyond the validation of partial models, no support was evident for an organisationally embedded validation mechanism which "cuts across" the partial models, drawing their concepts into a unified cognition. Both the suitability and comprehensibility of a technique are important factors in model validation, and it was evident in the survey that different integration strategies offered different advantages.

In classical techniques it was observed that triggering sources such as external events are used to trace the execution paths in models. In process-oriented techniques the resultant process execution sequence is determined implicitly in structural process models and explicitly in behavioural process models. The validation of execution paths is also advanced in behaviour process models through synchronisation and decision constructs. In OOA/D techniques, process sequences apply when object interactivity is high while object state sequences (in accordance with object lifecycles) apply when object interactivity is low. In business-oriented techniques, it was quite clear that organisational processing structures or organisation communication provide focal units for business processing.

A common theme for model validation is the notion of event. At discrete points within a timeline, it can be seen that an event triggers actors to execute business processes, through which information may be accessed, and further events, possibly invoking some processing in the environment (i.e. outside the business domain), may result. Intrinsic to the initial event is some defined purpose which motivates an execution path all the way to the final event. The final event signifies the (logical) termination of processing; i.e. representing the organisation's recognition that no further processing should proceed. Such an execution path is referred to as a *scenario*. For the purposes of model validation therefore, the following principle is proposed:

Principle – *Scenario Validation*

A technique should provide an explicit notion of scenario for model validation. □

Of issue is the organisational embedding of a scenario. An obvious choice is the notion of a *business transaction*, drawn from a Macroeconomics perspective of organisations [BM91]. Business transactions concern the exchange of (goods and) services using business transactions as the fundamental (accounting) unit. Clearly, the determination of a business transaction cost requires an understanding of the processing undertaken. It may be simple involving a low interactivity: a small set events and business processes. Or it may be complex involving a high-interactivity over a "long" temporal duration: a large set of events and business processes, typically involving multiple organisations.

At a first glance, it may seem that the choice of business service over business transaction as a scenario is arbitrary. Certainly, both terms tend to be used interchangeably in IS research. A cause-effect distinction however, leads to the preference of business transactions. That is, business services are a cause reflecting, more than anything else, client requirements of the organisation whereas business transactions are the effect; in effect, the scenario which results. In this sense, the concept of business transaction as an organisational artefact may be unified with the concept of workflow as an IS design and implementation artefact, i.e. *business transaction workflow*.

3.3 Service information hiding

A key distinction in business transactions is that between business services and business processes. This is, in fact, a generalisation of the distinction between events and processes. An event, afterall, is associated with some intention, and more than one event may share the same intention. In a business sense, intentions are denoted by business services.

As such, business services are a described (external) organisation of functionality which do not do anything as such, other than being associated with process interactions. This may involve external access such as client requests and responses from outside organisations, or internal access to services in different parts of an organisation. Inherently, they have a set of *states*, e.g. initiated, processing, rejected, and each state is associated with, and dependent on, a particular course of action resulting from a particular event. Business processes, on the other hand, are a prescribed (internal) organisation of functionality reflecting the mechanisms by which business services are delivered. Unlike services, they perform concrete actions, (e.g. data transformations, updates and retrievals) and their states are (relatively speaking) dependent on the success of their processing.

In classical modelling techniques, no explicit service concept exists. Business services are, therefore, dealt with in an arbitrary fashion. In process modelling techniques for instance, services are qualified (informally) in event descriptions and, therefore, directly trigger processes (see examples in [You89]).

Similarly in OOA/D techniques, events directly trigger methods in object types. Since business-oriented techniques adapt classical techniques, their treatment of business services is also arbitrary. For example in [Ram94], business services (functions) are related to business processes (tasks), but this is no more than a reference; i.e. triggering relates directly to business processes.

The common problem in these approaches is the *direct* triggering of processes (or methods) given the context of triggering. From the point of view of the environment or from different parts of an organisation, the actual business processes triggered for some business service request are inconsequential for the formulation of the request. That is, the request is issued for a business service and as a result some internal mechanism is used to determine what action to take. The implication is that when processes are reengineered, the actual request is not affected.

It may be recognised that the above paragraph reflects the *Information Hiding* principle for (software) module design which requires that internal aspects of modules be insulated from other modules. When applied to a technique's treatment of business service triggering, it can be seen that a service concept should explicitly be supported to coordinate the execution of number of processes and insulate them from service requests. Or in a more general form, the following principle results:

Principle – *Service Information Hiding*

A technique should allow the formulation of service requests to be independent of their actual processing. □

Figure 2 illustrates the ramifications for an inter-organisational workflow. Three organisations - a client, a server and a supplier (to the server) - are depicted with their set of services (small boxes) as interfaces to their processes (ellipses). The ordered set of arcs depicts messages to/from services carrying requests and responses. Within the services (somehow), the appropriate workflows are activated. Direct process to process triggering is allowed, provided the processes lie within the same service context, i.e. do not violate the *Service Information Hiding*.

Given the explicit incorporation of a service construct, it can be seen that workflows can be deployed in open distributed architectures (e.g. a Trader), and be globally accessed (potentially dynamically).

3.4 Cognitive sufficiency

The fourth problem is fairly general in its diagnosis. It addresses the missing constructs in techniques which yield collectively a sufficient cognition. Fundamental areas are described below.

Process modelling cognition

Although techniques allow either structural or behavioural aspects of processes to be modelled (given their distinctive purposes of analysis), both are required for capturing a business transaction's execution semantics. Structured process models, e.g. a data flow diagram (DFD) [You89], describe data flows, data stores and data transforming processes. Behavioural process models, e.g. Task Structures [HN93], describe process and their control flows.

Clearly, behavioural aspects are crucial for the specification of execution semantics. Structural concepts are also important for two reasons. Firstly, data flows are transmitted during process invocation. As

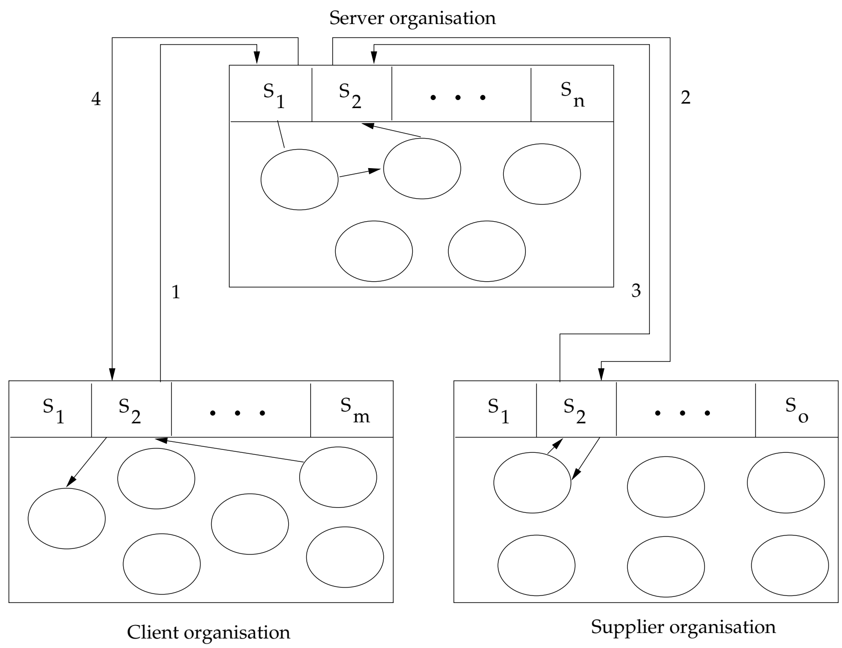

Figure 2: Operational business modelling for library example in [Ram94]

containers of data, they should be differentiated from the set of attributes they transmit. For example, a process precondition should make reference to a particular data flow type, when other data flow types may have the same set of attribute types as it. Secondly, process specifications would be incomplete without the incorporation of the data transformations and the data stores involved.

HCI

Another area of cognitive insufficiency is the conceptualisation of human to computer interaction (HCI) by techniques. Although traditionally determined at a detailed design stage, HCI *points* and their *dialogues* may be defined in a process model to reduce the *waterfall* between conceptual, design and implementation levels [DMHB90]. Moreover, it is possible to derive them from a business transaction's semantics; e.g. in [Ram94], if human and computerised actor types are involved in an action, that action is refined into a method of an external object type (a form).

Temporal aspects

Finally, the treatment of temporal aspects is partial despite the (obvious) impact of time in business domains. In Structured Analysis and other DFD techniques, a "clock" symbol is introduced to indicate time triggers on processes, e.g. end of the week. This however is only a comprehensibility feature. In [Ram94], time triggers are incorporated into the event specification language for object types, thereby extending the concern to expressive power. Further to expressive power, the need for time functions and variables are obviously required. A survey of techniques (including TEMPORA) dealing with time [TL91], demonstrates the incorporation of time on action preconditions, and relatedly dynamic constraints on object type specifications.

It is clear that these incorporations of temporal aspects relate to the specification of process preconditions. Of course, time should also be important in postconditions, e.g. as motivated by event scheduling requirements in [Bri90]. This, of course, further qualifies process dependency since a number of processes may be required to execute at some time, simultaneously (parallelism) or within a time duration of each other (sequence). Also a process may execute repeatedly within a certain time period (repetition), until some condition is satisfied. It is interesting to note that some model-based formal specification languages e.g. PAISley [Zav86], algebraic specification languages e.g. Real-Time Process Algebra [BB91], and Petri net based approaches e.g. ExSpect [HSV89], all address temporal aspects in process specifications. Indeed, the traditional domain for this consideration has been in real-time systems. The need of temporal specification, is nonetheless, evident in business domains. For example in law courts, matters are scheduled and adjourned at designated times, and in doing so, the availability of required documentation is requested (to outside organisations and parties) within a certain duration prior to the court hearing.

As a result of these areas of deficiency, the following principle is proposed:

Principle – *Cognitive Sufficiency*

A technique should provide a sufficient cognition of a model such that the need for fundamental business process execution assumptions is eliminated. □

3.5 Execution resilience

The fifth problem relates to error handling at the conceptual level. Of course at the conceptual level, database constraints and process pre- and post-conditions define an error free IS state. Moreover, specifications may be verified to eliminate erroneous specifications. However, *operational* errors can still occur beyond the control of an IS. For example, in a library domain, the existence of an overdue item (over different categories of duration) is an operational error caused by a borrower. Similarly, a missing enrollment confirmation in the required time since an initial enrollment, is caused by a student. Finally, a system crash is one of several examples of operational errors resulting in non-deterministic processing failures.

A sound conceptualisation of business transaction semantics requires the inclusion of operational error handling. However, no evidence in the assessed techniques indicated an explicit consideration of this. In most cases, an expressive power is available for dealing with errors like the first two examples. In process-centric techniques for example, error handling processes may be defined to scan a database for problematic object states (overdue item, unconfirmed enrollment). In state-centric techniques, an error handling process may be fired as a result of an object's transition into a problematic state.

The treatment of non-deterministic failures has recently become the subject of workflow implementation specifications. Under the traditional model, a transaction binds a set of database operations into an atomic unit of execution. Following the requirement of *failure atomicity*, a transaction's changes to a database(s) are *committed* if the execution is successful or *rolled-back* if not. Following the requirement of *execution-atomicity*, the concurrent execution of transactions should have the same effect as if they were executed in a *serialisable* order.

Workflows are more complex structures than traditional transactions, and it is unacceptable that the failure of any one of its processes results in the rollback of the entire workflow. In particular, when a process fails, an *undo* possibly through the execution of some other process, i.e. a *compensation*, or a *redo* possibly through a contingent process, should follow. Compensations are necessary since a process can commit and release its resources prior to a workflow reaching a termination state, therefore allowing other processes to access its updates. From this discussion, it can be seen that designers should be allowed to specify failure atomicity. Execution atomicity specifications which circumvent normal transaction behaviour should also be allowed; e.g. for different tasks updating logically related objects in a federated set of databases (geographically distributed), which for efficiency sake, commit and release locks individually. This means that the updates may be "seen" by concurrently executing database transactions. If any one of these updating tasks fails, compensation to "undo" the commits are required.

The lack of explicit support for operational error handling is indicative of the traditional IS development approach which prefers this treatment at the implementation level. A certain amount of this detail is, afterall, implementation-oriented (e.g. checking for a DBMS-defined deadlock error code after an update to determine whether a retry should be issued). Yet it should be recognised that operational error handling is an inherent part of a business transaction's execution semantics; i.e. its *execution resilience* semantics.

Significantly, the support of model execution in a large number of techniques, means that execution resilience can actually be validated and "tested" at the conceptual level. Given the disjointness (and complexity) of operational error handling - best presented as a separate layer to preserve a model's general cognition - model execution is considered critical. Therefore the following principle results:

Principle – *Execution Resilience*

A technique should support the handling of operational errors, so that a resilient execution of the conceptual model results. □

4 Conclusion

Conceptual modelling techniques haven proven to be invaluable for early and critical phases of analysis and design. When "bundled" into well-formed IS methods, a navigation from organisational analysis to implementation specifications is possible. For an essential problem-solving insight into different aspects of business domains, techniques should demonstrate a conceptualisation, expressive power, comprehensibility, formal foundation and business suitability. In the last of these aspects, the absence of a universal organisational theory has meant that techniques can only increase in technical effectiveness as newer insights are obtained from practical experience, and as the functionality of IS platforms expands. This makes it difficult to assess the adequacy of techniques to support a sound conceptualisation of business domains.

In this paper, the business suitability of workflow modelling was diagnosed for that class of business processing which is mission-critical in nature and which is amenable to strict operational procedure. This was based on an understanding of the approaches and (in)adequacies of a number of integrated techniques. Five business suitability principles were proposed: organisational embedding, scenario validation, service information hiding, cognitive sufficiency and execution resilience. The major benefit of these lie in assessing techniques, whereby immediate attention may be drawn to the general areas of deficiency we encountered. To overcome these and other deficiencies, it should now be clear that enhancing a technique requires a simultaneous consideration of all the conceptual modelling requirements. Given the large number of useful contributions in the field, we advocate a synthetic rather than analytic approach to developing techniques. Another benefit lies in the insight gained for workflows involving global access and inter-organisational collaboration. A clear separation between business service, business process and business transaction is essential for such specifications.

Much of this paper describes the vision for the development of a conceptual modelling kernel which we are specialising for workflow specifications. Currently we are applying techniques of the kernel to a real-world case study. Future work will focus on the formal semantics and model verification.

References

[AMP94] A.I. Anton, W.M. McCraken, and C. Potts. Goal decomposition and scenario analysis in business process reengineering. In G. Wijers, S. Brinkkemper, and I. Wasserman, editors, *Proceedings of the Sixth International Conference CAiSE'94 on Advanced Information Systems Engineering*, volume 811 of *Lecture Notes in Computer Science*, pages 94–104, Utrecht, The Netherlands, June 1994. Springer-Verlag.

[BB91] J.C.M. Baeten and J.A. Bergstra. Real Time Process Algebra. *Formal Aspects of Computing*, 3:142–188, 1991.

[BB95] A.T. Berztiss and J.A. Bubenko. A software process model for business reengineering. In *Proceedings of Information Systems Development for Decentralized Organizations (ISDO95), an IFIP 8.1 Working Conference*, pages 184–200, Trondheim, Norway, August 1995. Chapman & Hall.

[BHPC96] A.P. Barros, A.H.M. ter Hofstede, H.A. Proper, and P.N. Creasy. Business Suitability Principles for Workflow Modelling. Technical Report 380, Department of Computer Science, University of Queensland, Brisbane, Australia, August 1996.

[BM91] W. Boyes and M. Melvin. *Macroeconomics*. Houghton Mifflin, Boston, Massachusetts, 1991.

[Boo91] G. Booch. *Object-Oriented Design with Applications*. Benjamin Cummings, Redwood City, California, 1991.

[Bri90] S. Brinkkemper. *Formalisation of Information Systems Modelling*. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, 1990.

[Bro87] F.P. Brooks Jr. No silver bullet: essence and accidents of software engineering. *IEEE Computer*, 20(4):10–19, April 1987.

[BS87] D. Benyon and S. Skidmore. Towards a Tool Kit for the Systems Analyst. *The Computer Journal*, 30(1):2–7, 1987.

[CY90] P. Coad and E. Yourdon. *Object-Oriented Analysis*. Yourdon Press, New York, New York, 1990.

[DB91] R.C.J. Dur and P.W.G. Bots. Dynamic Modelling of Organizations Using Task/Actor Simulation. In R.L. Crosslin and H.G. Sol, editors, *Proceedings of the Second International Working Conference on Dynamic Modelling of Information Systems*, pages 49–71, Amsterdam, The Netherlands, 1991. Elsevier Science Publishers.

[DeM78] T. DeMarco. *Structured Analysis and System Specification*. Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

[Die94] J. Dietz. Business modelling for business redesign. In *Proceedings of the 27th International Conference on System Sciences*, pages 723–732, Honolulu, Hawaii, August 1994.

[DMHB90] E.J.T. van Dinter, M.P.W. Martens, A.H.M. ter Hofstede, and S. Brinkkemper. A Support Tool for Integrated Dialogue Specification. In R. Norman and R. van Gendt, editors, *Proceedings of the Fourth International Workshop on CASE*, pages 290–301, Irvine, California, December 1990. IEEE Computer Science Press.

[DO85] G.B. Davis and M.H. Olson. *Management Information Systems: Conceptual Foundations, Structure and Development*. McGraw-Hill, New York, New York, 1985.

[DP95] V. De Antonellis and B. Pernici. Reusing specifications through refinement levels. *Data & Knowledge Engineering*, 15(2):109–133, April 1995.

[FHL+98] E.D. Falkenberg, W. Hesse, P. Lindgreen, B.E. Nilsson, J.L.H. Oei, C. Rolland, R.K. Stamper, F.J.M. Van Assche, A.A. Verrijn-Stuart, and K. Voss, editors. *A Framework of Information Systems Concepts*. IFIP WG 8.1 Task Group FRISCO, 1998.

[FL80] F. Flores and J.J. Ludlow. Doing and Speaking in the Office. In *Decision Support Systems: Issues and Challenges*. Pergamon, 1980.

[FYW94] G. Fitzpatrick, Y. Yang, and J. Welsh. Supporting Cooperative Work Processes - A Survey of Systems and Issues. Technical Report 304, Department of Computer Science, University of Queensland, Brisbane, Australia, June 1994.

[GHS95] D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modelling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, April 1995.

[GS86] C. Gane and T. Sarson. *Structured System Analysis: Tools and techniques*. IST Databooks. MacDonald Douglas Corporation, St. Louis, 1986.

[HN93] A.H.M. ter Hofstede and E.R. Nieuwland. Task structure semantics through process algebra. *Software Engineering Journal*, 8(1):14–20, January 1993.

[HPW93] A.H.M. ter Hofstede, H.A. Proper, and Th.P. van der Weide. Formal definition of a conceptual language for the description and manipulation of information models. *Information Systems*, 18(7):489–523, October 1993.

[HSV89] K.M. van Hee, L.J. Somers, and M. Voorhoeve. Executable Specifications for Distributed Information Systems. In E.D. Falkenberg and P. Lindgreen, editors, *Information System Concepts: An In-depth Analysis*, pages 139–156. North-Holland/IFIP, Amsterdam, The Netherlands, 1989.

[ISO87] *Information processing systems – Concepts and Terminology for the Conceptual Schema and the Information Base*, 1987. ISO/TR 9007:1987.

[JCJO92] I. Jacobson, M. Christerson, M. Jonsson, and P. van Overgaard. *OO Software Engineering, A Use Case Driven Approach*. Addison-Wesley, Reading, Massachusets, 1992.

[Kun93] D.C. Kung. The Behaviour Network Model for Conceptual Information Modelling. *Information Systems*, 18(1):1–21, 1993.

[LGN81] M. Lundeberg, G. Goldkuhl, and A. Nilsson. *Information Systems Development - A Systematic Approach*. Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[LK95] P. Loucopoulos and E. Kavakli. Enterprise modelling and the teleological process design and database design. *International Journal of Cooperative Information Systems*, 4(1):45–79, January 1995.

[ML83] J.L. Malouin and M. Laundry. The Mirage of Universal Methods in Systems Design. *Journal of Applied Systems Analysis*, 10:47–62, 1983.

[MMNR90] V.J. Mercurio, B.F. Meyers, A.M. Nisbet, and G. Radin. AD/CYCLE strategy and architecture. *IBM Systems Journal*, 28:170–187, 1990.

[Ram94] G.J. Ramackers. *Integrated Object Modelling, an Executable Specification Framework for Business Analysis and Information System Design*. PhD thesis, University of Leiden, Leiden, The Netherlands, 1994.

[RBP+91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson. *Object-Oriented Modeling and Design*. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[RD82] G. Richter and R. Durchholz. IML-Inscribed High-Level Petri Nets. In T.W. Olle, H.G. Sol, and A.A. Verrijn-Stuart, editors, *Information Systems Design Methodologies: A Comparative Review*, pages 335–368. North-Holland/IFIP WG8.1, Amsterdam, The Netherlands, EU, 1982.

[Rod91] T. Rodden. A survey of CSCW systems. *Interacting with Computers*, 3(3):319–353, 1991.

[SK86] A. Sølvberg and C.H. Kung. On Structural and Behavioural Modelling of Reality. In T.B. Steel Jr and R. Meersman, editors, *Database Semantics (DS-1)*, pages 145–171. North-Holland/IFIP, Amsterdam, The Netherlands, 1986.

[SM88] S. Shlaer and S.J. Mellor. *Object-Oriented Systems Analysis: Modeling the World in Data*. Yourdon Press, New York, New York, 1988.

[TL91] C.I. Theodoulidis and P. Loucopoulos. The Time Dimension in Conceptual Modelling. *Information Systems*, 16(3):273–300, 1991.

[WW93] D.G. Wastell and P. White. Using process technology to support cooperative work: Prospects and design issues. In D. Daiper and C. Sanger, editors, *CSCW in Practice: An Introduction and Case Studies*, pages 105–126. Springer-Verlag, London, United Kingdom, 1993.

[You89] E. Yourdon. *Modern Structured Analysis*. Printice-Hall, Englewood Cliffs, New Jersey, 1989.

[Zav86] P. Zave. Salient Features of an Executable Specification Language and Its Environment. *IEEE Transactions on Software Engineering*, 12(2):312–325, 1986.