December 1997

Behavioural Issues in Information Systems Design, Development and Implementation: A Process Modelling Framework

Gerard McGrath
Macquarie University

Behavioural Issues in Information Systems Design, Development and Implementation: A Process Modelling Framework

G.M. McGrath
Joint Research Centre for Advanced Systems Engineering, Macquarie University, Sydney, Australia
Tel: +61 (0) 2 9850 9581
Fax: +61 (0) 2 9850 9102
E-mail: mmcgrath@macadam.mpce.mq.edu.au

Abstract
A software engineering process modelling framework is presented. The focus of this work is on behavioural aspects - at the individual, project team, corporate-wide and external environment levels. The aim is to empower project managers and others involved in information systems development to better deal with the critical human aspects inherent in all non-trivial projects. A feature of the framework is that user views, in a variety of modelling formalisms, may be extracted from a common, highly-abstracted conceptual model. Examples of "user views" represented and implemented in Prolog and within an expert system shell are presented. This work extends previous research into the derivation and automated implementation of formal models of organisation and management theories.

Executive Summary
Most software engineering process modelling work has focused on the evolutionary behaviour of artefacts through the various developmental stages rather than on the behaviour of those creating the artefacts. This restricted approach ignores the fact that the ever expanding range of computing and communications technologies plays only an enabling role in the development of the systems eventually used to achieve a user's defined business objectives. The building of information systems involves, at its core, the still emerging discipline of software engineering as well as other more traditional areas of engineering and science and a number of non-technical management, social and organisational disciplines. In essence, systems are not built in a vacuum, but within organisational environments where outcomes are heavily influenced by a myriad variety of internal and external socio-technical factors. The organisation management and social science literature is rich in descriptions of models relevant to the software development process. In general, however, even the most prescriptive of these models are not specified with anything like the formality required to permit convenient translation to an automated implementation. Nevertheless, key aspects of these models can be represented using more formal modelling approaches. This, in turn, allows the definition and development of evaluation environments that allow the rapid prototyping of persistent process models which effectively embody technical, organisational, social and human factors.

In this paper, a software engineering process modelling framework is presented. The focus of this work is very much on behavioural aspects - at the individual, project team, corporate-wide and external environment levels. A feature of the framework is that user views, in a variety of modelling formalisms, may be extracted from a common, highly-abstracted conceptual model (represented in entity-relationship form). This is consistent with a growing consensus that (more formal) modelling of organisation and management theory (OMT) requires a "horses for courses" approach. That is, different theories are often best represented using different paradigms, with the conceptual model serving as a common reference point through which consistent external views may be derived and verified. The utitlity of the process modelling framework is demonstrated through informal mappings of organisational power concepts into models implemented: 1) in Prolog; and 2) within an expert system shell.

Introduction

In recent years an enormous amount of research has been devoted to process modelling and related areas, but industry experience with process improvement initiatives has been disappointing [1]. Patricia Sachs (1995) has convincingly argued that many business process reengineering (BPR) failures have resulted from too narrow a focus and that a more holistic approach is required, In particular, she argues that *"business process reengineering --- fails to acknowledge how work is carried out and the part that human ingenuity plays in it"* and that *"(1) if only the organizational (explicit structures) of work are considered in designing work and (2) the importance of learning is left out, there will be negative consequences in the conception and implementation of* [process] *design"* (pp. 38-39).

Similarly, as noted by Curtis, Krasner and Iscoe (1988), most software engineering process modelling work has focused on the evolutionary behaviour of artefacts through the various developmental stages rather than on the behaviour of those creating the artefacts. This restricted approach ignores the fact that the ever expanding range of computing and communications technologies plays only an enabling role in the development of the systems eventually used to achieve a user's defined business objectives. The building of information systems involves, at its core, the still emerging discipline of software engineering as well as other more traditional areas of engineering and science and a number of non-technical management, social and organisational disciplines. In essence, systems are not built in a vacuum, but within organisational environments where outcomes are heavily influenced by a myriad variety of internal and external socio-technical factors. The organisation management and social science literature is rich in descriptions of models relevant to the software development process. In general, however, even the most prescriptive of these models are not specified with anything like the formality required to permit convenient translation to an automated implementation. Nevertheless, key aspects of these models can be represented using more formal modelling approaches. This, in turn, allows the definition and development of evaluation environments that allow the rapid prototyping of persistent process models which effectively embody technical, organisational, social and human factors.

In this paper, a software engineering process modelling framework is presented. The focus of this work is very much on behavioural aspects - at the individual, project team, corporate-wide and external environment levels. A feature of the framework is that user views, in a variety of modelling formalisms, may be extracted from a common, highly-abstracted conceptual model (represented in entity-relationship form). This is consistent with a growing consensus that (more formal) modelling of organisation and management theory (OMT) requires a "horses for courses" approach (Masuch 1992). That is, different theories are often best represented using different paradigms, with the conceptual model serving as a common reference point through which consistent external views may be derived and verified. The utitlity of the process modelling framework is demonstrated through informal mappings of organisational power concepts into models implemented: 1) in Prolog; and 2) within an expert system shell.

Representations Of Organisation And Management Theory

In an important recent work, Jaques (1996) has argued that, despite massive efforts by organisation and social science experts, only the merest beginnings of an organisation and management science have become evident. In arguing this case, he reserves particular criticism for *"the pile of vague and ill-defined terms that litter the field"* and notes that *"Without --- clear meaning it is impossible to think, or to test propositions, or to talk to one another with any hope of understanding"* (p. 10). These observations are consistent with an increasing body of OMT research has addressed the benefits of formal models as opposed to informal, literary theorising. For example, Bendor and Moe (1992) claim that more formal representations clarify chains of reasoning and simplify the task of verifying that conclusions do indeed follow from assumptions; McGrath (1994) points to the clarification of concept overlaps, ambiguities and inconsistencies; and Curtis, Kellner and Over (1992) suggest that the properties of multiple model perspectives can be analysed more accurately where representation constructs are formally constrained. In addition, formal models can generally be implemented more readily and modelling both "soft" and "hard" data using a common approach reduces the possibility that the critical softer aspects will be overlooked. There is, however, little consensus on the merits or otherwise of the various candidate specification languages and schemes.

---

[1] In a 1994 survey of 350 US companies that had undertaken BPR exercises, only 17% of the companies surveyed expressed satisfaction with the results *(The Australian, 28/6/94)*.

Cecez-Kecmanovic and Marjanovic (1995) have proposed a theoretical framework for investigating the nature of organisational processes from a social interaction perspective. A detailed description of this promising (but relatively immature) modelling perspective is beyond the scope of this paper but they, along with others are critical of the well-established, more formal alternatives employed for the representation and communication of organisational knowledge: specifically, the management information systems (MIS), rational choice (RC), artificial intelligence (AI) and object-oriented (OO) approaches. Below, some brief observations relevant to the AI and RC approaches are offered.

Organisations can be viewed as information processing systems (IPS). Newell and Simon (1972) presented an IPS model in which organisational parties, with limited information processing capabilities, can solve complex problems through *division of labour, specisalisation* and *cooperation.* This mirrors a common AI problem solving method, which involves: the recursive dissection of a problem into subproblems (division of labour); the selection of subproblem-appropriate operators at each node of the solution search tree (specialisation); and backtracking, combined with directed information flows to promising processes (cooperation). The analogy can be taken further, since the organisational rules used to resolve routine decision making situations are often expressed in "Standard Operating Procedure" manuals in a form very close to expert system, *If-Then* style production rules (or, at least, can be readily and naturally translated to this form). Thus, the attractiveness of the AI approach for modelling and simulating many organisational decision making situations can readily be seen.

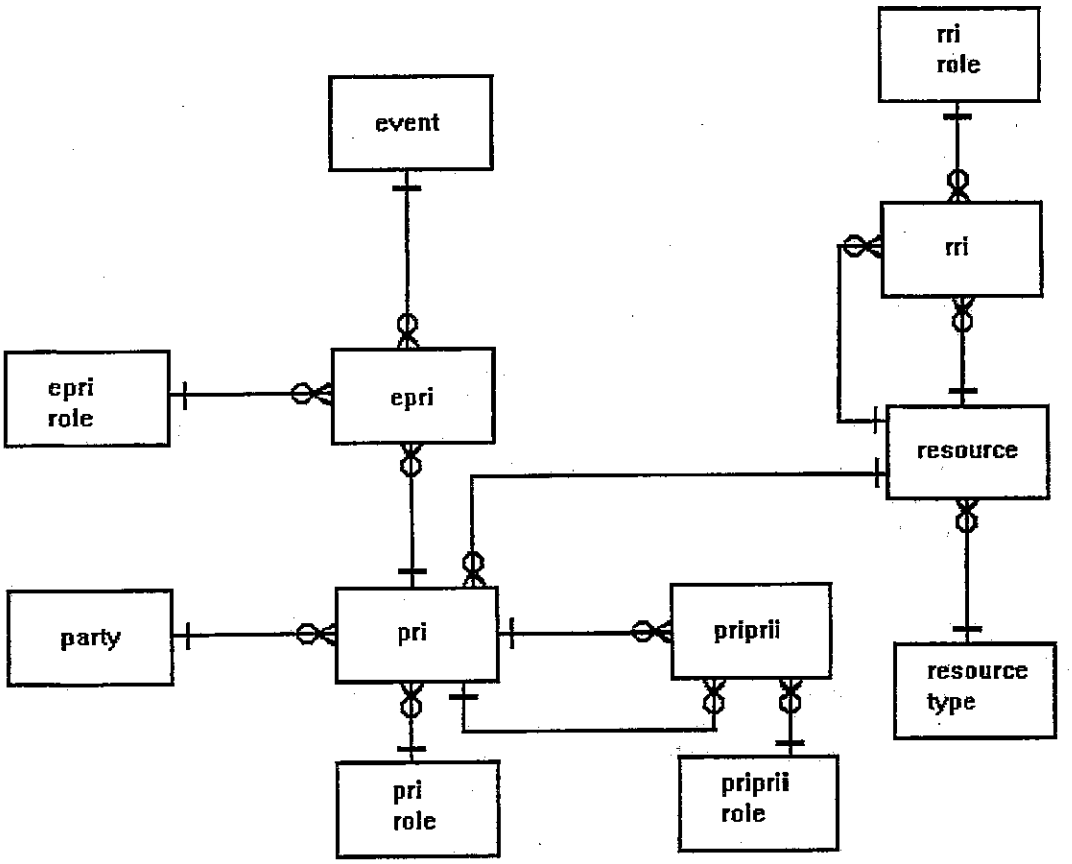
Bendor and Moe (1992) while noting that, within the subfield of studies of public organisations, RC analysis has become the dominant mode of thinking, point to a number of problems. Specifically: 1) some heroic assumptions are made regarding the information processing capabilities of organisational parties; 2) little allowance is made for "out-of-equilibrium" behaviour; 3) there is an undue focus on dyadic relations (e.g. an agency and a median legislator); and 4) the goals of key parties are often treated in a highly-stylised manner (almost to the point of caricature). Despite this, they point to the benefits and successes of the approach, together with the speed with which the paradigm has been adopted (particularly within the subfield of the study of public organisations). The authors conclude that the gap between the rich literary descriptions of organisations (employed for the great majority of OMT specifications) and the spare, quantitative RC models is so wide that researchers might conclude that they face a dichotomous choice: either pursue a descriptive approach, resulting in rich but loosely-structured models, or give up most of what they know and build rigorous but highly-simplified models!

Recently, however, considerable attention has been focused on the relative merits of *qualitative* and *quantitative* representations of organisations and OMT. Early models were highly quantitative: principally because, as noted by Masuch (1992), if a computerised implementation was desired, model developers had little alternative. Masuch though, further notes that much OMT is highly-qualitative and this, combined with the relatively recent proliferation of powerful AI tools outside laboratories, has resulted in a pronounced shift towards qualitative models. Despite this trend, however, he cautions against over-enthusiasm for the qualitative approach and quite sensibly suggests that a quantitative conceptualisation scheme should always be employed where details of the phenomenon under study are most naturally expressed in quantitative terms. This accords with the process modelling philosophy underpinning this work where, despite employing a common approach at the highly-abstracted conceptual level, user views (mappings from the conceptual model) may be essentially qualitative, quantitative or a mixture of both. This is consistent with the view of Curtis et al. (1992) who have argued that different modelling objectives, user diversity, conflicting requirements and the need for both large and small-grained levels of abstraction all demand OMT models permitting multi-paradigm representations.

An Organisational Process Modelling Framework

The framework has three levels: with the universe of discourse (UoD) at Level 1; the conceptual model at Level 2; and the external models (or user views) at Level 3. In developing this framework the work of ISO Technical Committee 97, in defining the foundations of the 3-schema database management system architecture (van Griethuysen 1982), has been drawn upon. Thus the *UoD* refers to *that collection of objects, from a real or postulated world, that is being described.* It should be noted that, while the world of interest here is centred on software design and development processes (and, more particularly, on behavioural aspects of these processes), the framework is applicable to organisational processes in general.

The UoD representation is based on the layered behavioral model of Curtis et al. (1988). The *tech/admin* layer at its core refers to the parties, artefacts, roles etc. that form the basis of most process models; outside that, the *individual* layer refers to cognitive aspects (attitudes, opinions, beliefs, knowledge etc.); next, the *project team* layer focuses mainly on group dynamics; the *company* layer is concerned with corporate-wide aspects, such as organisational power, politics, culture and structure; and, finally, the *business milieu* layer refers to an organisation's interaction with its external environment.



Glossary:

pri	**party-resource involvement**
epri	**event-party-resource involvement**
priprii	**pri-pri involvement**
rri	**resource-resource involvement**

Figure 1: A Simplfied, Abstract Conceptual Process Model using "Information Engineering" Conventions (Finkelstein, 1989).

At the second level, the *conceptual model* defines *the objects of the UoD, including rules governing allowable classifications, states, transitions and constraints* (van Griethuysen 1982). Part of the conceptual model is illustrated in Figure 1. The model is represented in entity-relationship form, it is highly abstracted and it is a *common denominator* schema, as defined by Curtis et al. (1992). That is, it is a schema that specifies only the essentials of the vital or core processes, leaving aside any external representation considerations. The entity-relationship approach has been employed not because it is unarguably superior to competing formalisms but because: 1) it is probably the most popular and best known data modelling approach used within the information systems arena; 2) there is a well-defined abstraction process for entity-relationship models that employs much the same "super" entity types used within process modelling; and 3) the whole rationale for the development of the entity-relationship modelling approach was to provide a means of unifying competing (information) modelling formalisms.

At Level 3, *external models* or *user views* are *mappings from all or part of a conceptual model to a language or representational form of a user's choosing* (van Griethuysen 1982). Examples of two such mappings will now be informally introduced. The examples are concerned with organisational change and, more specifically, with power source redistributions brought about by the development and implementation of new information systems (Markus 1983, Pfeffer 1981 and 1992).

resource	res-id	res-type
	sys101	system
	sys151	system
	sys-owner	ps
	sys-maintainer	ps
	expert-kn	ps
	info-provn	ps
	decn-making	ps
	surface-level-power	ps

party	party-id
	'Finance'
	'Team A'
	'Jill'
	'Pat'

pri	party-id	res-id	pri-role
	'Finance'	sys101	sys-owner
	'Team A'	sys101	sys-maintainer
	'Jill'	sys151	leader
	'Team A"	sys151	members

epri	event-id	party-id	res-id	epri-role
	'D1'	'Finance'	sys101	threatens
	'D1'	'Team A'	sys101	threatens

rri	res-id	res-id	rri-role
	surface-level-power	decn-making	derived-from
	decn-making	info-provn	derived-from
	info-provn	expert-kn	derived-from
	info-provn	sys-owner	derived-from
	expert-kn	sys-maintainer	derived-from

Table 1: Partial Relational View of Conceptual Model and Sample Data.

Example 1: Relational View

Entity-relationship to relational mapping is a well-understood and will not be described in detail here. This mapping, presented in Table 1, has mainly been included in order to introduce data employed in later examples.

Example 2: Logic-Based View

Relation intensions can serve as templates for Prolog assertions. Rarely, however, does this result in Prolog programs with clear declarative interpretations - thus negating one of the languages most important features. An alternative approach is illustrated in Figure 2. This approach relies on the fact that, particularly where abstraction is employed, entity-relationship models tend to be composed of collections of basic constructs of the one type and, further, that these structures are often compounded into superstructures which, again, are essentially of the same type.

The construct referred to is shown at the core of Figure 2. It depicts two entities[2] E_1 and E_2, involved in a relationship, R_1, with R_1r signifying the role the first entity plays in the relationship. This construct can be represented as the Prolog assertion template[3], Template 1:

isa($R_1r(E_1, E_2)$, 'R_1').

Extending now to the next layer in Figure 2, we have a compound structure which is of the same basic type as our core structure. To create the assertion template for this structure all that is required is: 1) to replace E_1, R_1r and 'R_1' in Template 1 by E_3, R_2r and 'R_2' respectively; and 2) to replace E_2 with Template 1's complete subject, $R_1r(E_1, E_2)$. Thus Template 2 is:

isa($R_2r(E_3, R_1r(E_1, E_2))$, '$R_2$').

[2] For the sake of brevity, and where there is no possibility of confusion, we shall henceforth refer to entity and relationship types simply as entities and relationships respectively.

[3] As with the de facto Edinburgh standard Prolog syntax, variables begin with capitals.

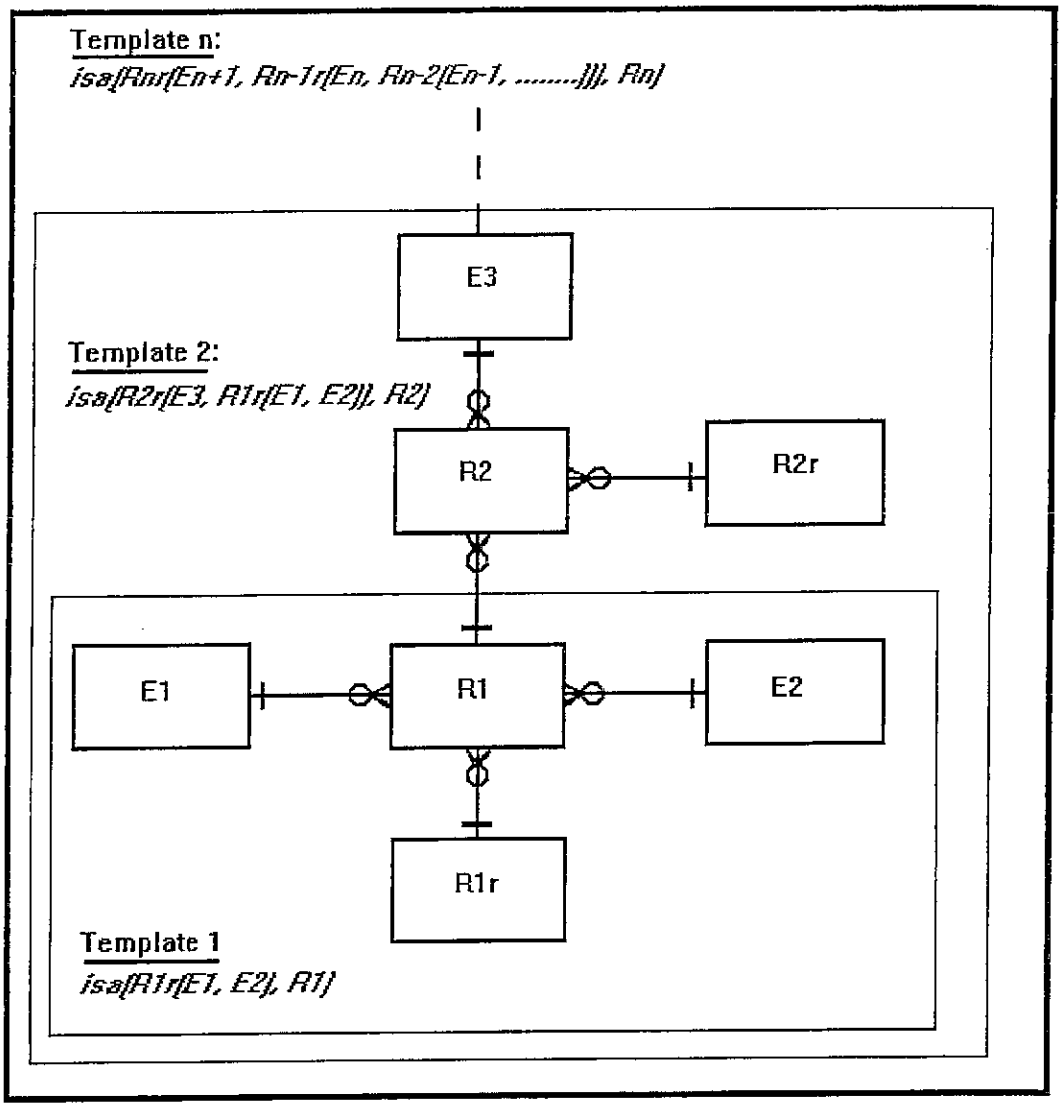


Figure 2: Entity-Relationship to Prolog Mapping Scheme.

This process can be extended indefinitely although, only very rarely would one need to move much beyond two levels. The reader should also note that mapping may begin at different points in the entity-relationship diagram and, at any point in the mapping process, the model can often be extended in a number of directions. As a result, there may be multiple assertion templates at each level.

Returning now to the conceptual process model in Figure 1, one starting point is at *pri.* This results in the assertion template:

isa(Pri_role(Party, Resource), pri)

instances of which are:

isa(sys_owner('Finance', sys101), pri)
and
isa(sys_maintainer('Team A', sys101), pri)

which have the declarative interpretations: 1) *(the party) Finance is the system owner of (the system) sys101 is a party-resource involvement* and 2) *(the party) Team A is the system maintainer of (the system) sys101 is a party-resource involvement.* Extending now to the next layer, we get the template:

isa(Epri_role(Event, Pri_role(Party, Resource)), epri)

and an instance of this assertion template is

isa(threatens('D1', sys_owner('Finance', sys101)), epri).

Not all entity-relationship constructs will conform to the multi-level template structure described above. Exceptions can be handled using an alternative approach detailed by Kowalski (1979). This involves breaking each n-ary relation down into n binary assertion templates. This is accomplished by: 1) declaring the existence of the relation through an *isa* assertion; and 2) creating an assertion template for each relation attribute (other than the primary key), with the attribute name as predicate and the primary key as subject. For example, the *resource* relation translates to:

isa(Res_id, resource)
and
res_type(Res_id, X).

One role of an external model is to provide a means of deriving new information from core conceptual data. With Prolog as the external model implementation platform, this mainly involves combining assertions, developed during the mapping processes described, into appropriate sets of rule conditions. Thus, the rule:

has_a(Party, derived_from(ps, Resource)) :-
(isa(sys_owner(Party, Resource), pri)
OR isa(sys_maintainer(Party, Resource), pri),
res_type(Resource, system)

may be used to determine all parties with power sources derived from direct involvement with information systems. With the sample data in Table 1, the results returned will be that both *Finance* and *Team A* have power (derived from ownership and maintenance respectively of *sys101).*

Example 3: Flex Expert System View

Flex™ (LPA 1992) is an expert system shell. It allows users to develop systems using a combination of frames and production rules (employing forward chaining, backward chaining or both).

The frames paradigm, developed in the early 1970s, as an alternative to the predominant production rule, functional programming and logic-based AI development approaches (Minsky 1975), is, essentially, an example of a very early object-oriented development method. Thus, a simple means of mapping from our conceptual model to a frame-based external view is to: 1) declare each entity and relationship type as a frame (object class); 2) declare each attribute as a frame slot; and 3) declare each entity and relationship type instance as a frame instance (object). Hence, in the Flex view, each row tuple in Table 1 becomes an instance of the object class (or frame) corresponding to its table name. Some examples of frame declarations follow:

frame pri;
default party_id is unknown and
default res_id is unknown and
default pri_role is unknown.

frame rri;
default res_id is unknown and
default res_id' is unknown and
default rri_role is unknown.

As with Prolog rules, expert system production rules are used in external models to derive new information from core conceptual data. The conditions of these rules are all expressed in terms of frames and their slots. However, additional frames, which serve as templates for holding the results of rule derivations, may be defined. In the example presented below, the aim is to determine all power

sources derived from parties' stakeholdings in information systems. We, therefore, define the following *power_source* frame:

```
frame power_source;
    default ps is unknown.
```

Pfeffer (1981, p.7) defines power as *"a force, a store of potential influence through which events can be affected"*. He describes power as *"a property of the system at rest"* and politics as *"the study of power in action"*. Pfeffer's stores of influence are *power sources* (examples of which are control over information flows, position in the communications network and expert knowledge) Power sources are classified generically as resources (see Table 1). Many organisation decisions and activities may result in a perceived and/or real redistribution of power sources. There are winners and losers, and losers may resist change. Markus (1983) has extended Pfeffer's work by proposing a power source distribution model and using it to (retrospectively) explain resistance to the implementation of an integrated financial information system in a large organisation. In a later work, Markus and Bjorn-Andersen (1987) have observed that the integration of isolated systems, in particular, is a task fraught with political difficulties. Specifically, they note that because the integration task cuts across organisational boundaries, changes in work-flow, communication patterns and control processes may lead to a significant shift in the organisational balance of power. In an earlier, related work (McGrath 1994), Markus's central concept (that systems integration leads to a power shift and consequential resistance) has been extended and implemented as an advisory expert system. The example presented here is based on this research.

Mackenzie (1986), in questioning the practical usefulness of the mutual dependence concept fundamental to much early research into the power model, has proposed a more complex model based on a "structured cascading of uncertainty and dependency". An implication of this is that, unless links between organisation parties, processes, artefacts and resources are first established, a change agent will face considerable difficulty in assessing the potential impact of any initiative. Thus, provided these linkages, plus relationships between power sources are defined, both the more obvious areas of potential resistance and consequential resistance may be predicted. This is important, because many organisation actors will jealously guard some power sources, yet be relatively unconcerned about others (Kotter and Schlesinger 1979). Knowledge of these factors is essential if a change agent is to choose appropriate tactics.

Earlier, a logic rule for deriving power sources derived from direct involvement with information systems was presented. This rule may now be rewritten in Flex as:

```
rule ps_1
    if PRI is some pri
        with party_id of Party and res_id of Resource and pri_role of PRI_Role
    and the res_type of Resource is system
    and PRI_Role is {sys_owner or sys_maintainer}
    then PS is a new power_source
        with ps of PRI_Role(Party, System)
```

which (loosely stated) declares that, within the information systems/software engineering domain, systems ownership and maintenance are basic sources of power. With our sample data, the system will infer the information that *Finance* and *Team A* possess base power derived from their roles in systems ownership and maintenance respectively.

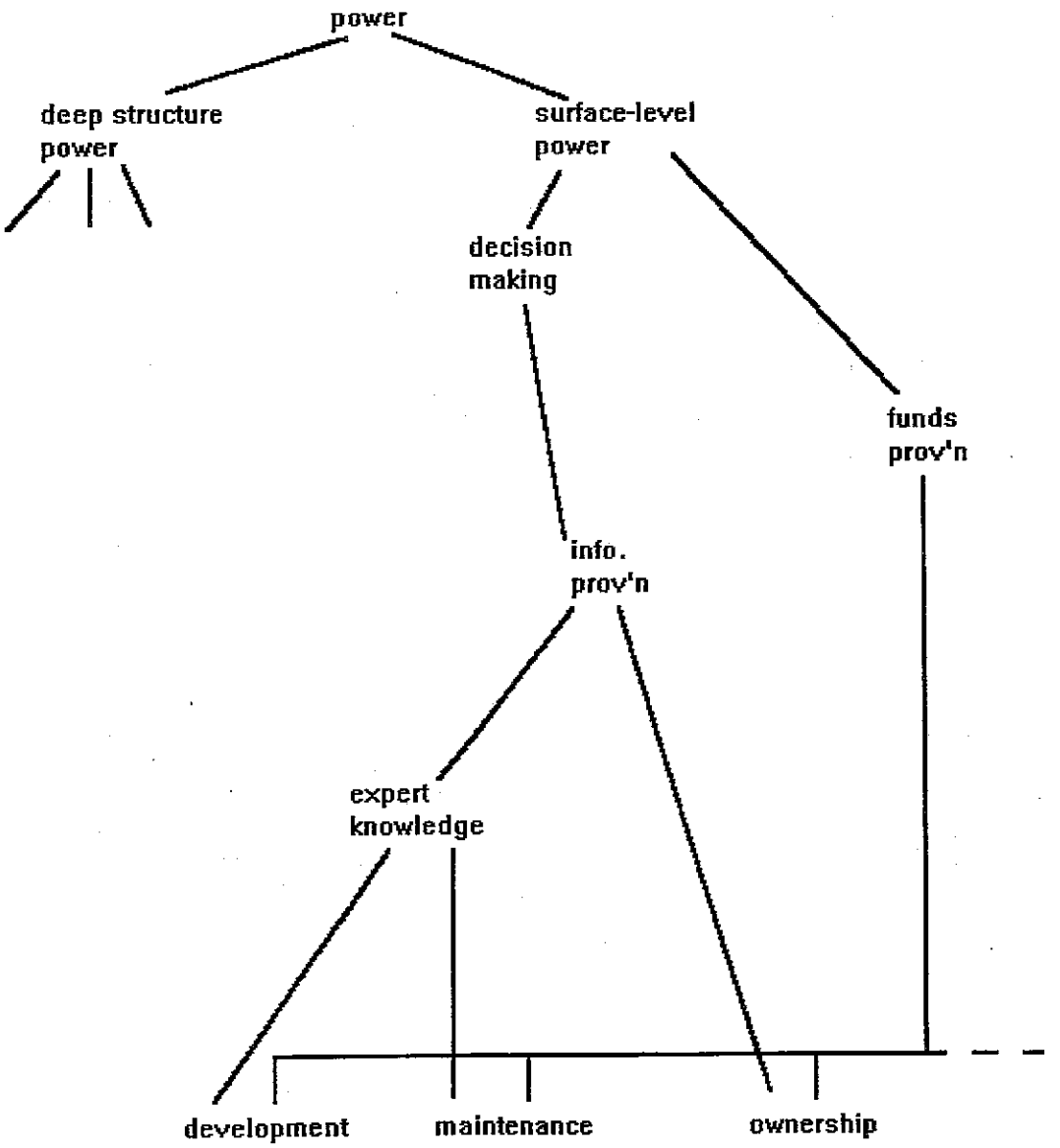


Figure 3: Partial Power Source Network.

Figure 3 displays a partial power source network developed during earlier research (McGrath 1994). "Consequent" power sources (i.e. power sources dependent on a direct information systems involvement) may now be derived through the additional production rule:

rule ps_2
if PS is some power_source
with ps of PRI_Role(Party, Resource)
and the res_type of Resource is system
and RRI is a rri
with rri_role of derived_from and res_id' of PRI_Role and res_id of Res_id
then PS' is a new power_source
with ps of Res_id(Party, Resource).

Through this rule, the system will report that both *Finance* and *Team A* have power derived from information provision and their ability to influence decision making processes and that, in addition, *Team A* have power derived from expert knowledge. The inference that these are all *surface-level* power sources (Frost 1987) is trivial (at least insofar as this example is concerned).

Further rules might now be added. These could be used to determine sources of likely resistance, based on threats to power sources resulting from events or decisions (e.g. the decision, *D1*, in Table 1). However, it must be emphasised that we are dealing here with *potential* resistance and that additional rules are required to assist the user in deciding whether resistance, based on threats to power sources is, in fact, likely.

Conclusion

The framework presented allows vital socio-technical and behavioural aspects to be incorporated into software engineering process models. The aim is to empower project managers and others involved in systems development to better deal with the critical human aspects inherent in all non-trivial projects.

This research is aimed at contributing to the development of a sound scientific base for software engineering process modelling and specifically, for that part of the development process concerned with behavioural issues. While a number of methods and techniques commonly employed in systems development work have reasonably well-developed scientific bases, the same can not be said of the total development process. Systems developed and implemented without a sound set of principles are doomed to unpredictability because no means exist to understand why projects succeed or indeed, why they fail. The consequence is that success can neither be sustained, nor failure avoided. The work presented in this paper is based on the premise that, by capturing and formalising relevant OMT concepts, we may provide a means of significantly improving the information systems development success rate.

6

References

Bendor, J. and Moe, T.M. "Bureaucracy and Subgovernments - A Simulation Model." In M. Masuch and M. Warglien (eds.), Artificial Intelligence in Organization and Management Theory, Amsterdam: North-Holland, 1992, pp. 119-141.

Cecez-Kecmanovic, D. and Marjanovic, O. "IT as a Medium for Social Interaction." Proceedings of the 6th Australasian Conference on Information Systems, Perth, Australia, Sept. 26-29, 1995, pp. 597-612.

Curtis, B., Krasner, H. and Iscoe, N. "A Field Study of the Software Design Process for Large Systems." Communications of the ACM, Volume 31, Number 11, 1988, pp. 1268-1287.

Curtis, B., Kellner, M.I. and Over, J. "Process Modelling." Communications of the ACM, Volume 35, Number 9, 1992, pp. 75-90.

Finkelstein, C. An Introduction to Information Engineering: From Strategic Planning to Information Systems. New York: Addison-Wesley, 1989.

Frost, P.J. "Power, Politics and Influence." In F.M. Jablin, L.L. Putnam, K.H. Roberts and L.W. Porter (eds.), Handbook of Organizational Communication: An Interdisciplinary Perspective, Beverly Hills: Sage, 1987, pp. 503-548.

van Griethuysen, J.J. Concepts and Terminology for the Conceptual Schema and the Information Base. ISO Technical Report ISO/TC97/SC5/WG3, 1982.

Jaques, E. Requisite Organization. (Revised second edition - Final Draft), Arlington, Virginia: Cason Hall and Co., 1996.

Kotter, J.P. and Schlesinger, L.A. "Choosing Strategies for Change." Harvard Business Review, Volume 57, Number 2, 1979, pp. 106-114.

Kowalski, R.A. Logic for Problem Solving. New York: North Holland, 1979.

LPA. flex Expert System Toolkit - Technical Reference. Logic Programming Associates Ltd, London, 1992.

McGrath, G.M. "MP/L1. Towards an Automated Model of Organisational Power." Proceedings of the 2nd IEEE Australian and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia, 29 Nov. - 2 Dec., 1994, pp. 487-491.

Mackenzie, K.D. "Virtual Positions and Power." Management Science, Volume 32, Number 5, 1986, pp. 622-642.

Markus, M.L. "Power, Politics and MIS Implementation." Communications of the ACM, Volume 26, Number 6, 1983, pp. 430-444.

Markus, M.L. and Bjorn-Andersen, N. "Power over Users: Its Exercise by System Professionals." Communications of the ACM, 1987, Volume 30, Number 6, pp. 498-504.

Masuch, M. "Introduction - Artificial Intelligence in Organisation and Management Theory." In M. Masuch and M. Warglien (eds.), Artificial Intelligence in Organization and Management Theory, Amsterdam: North-Holland, 1992, pp. 1-19.

Minsky, M. "A Framework for Representing Knowledge." In P. Winston (ed.), The Psychology of Computer Vision, New York: McGraw-Hill, 1975, pp. 211-280.

Newell, A. and Simon, H.A. Human Problem Solving. Englewood Cliffs, New Jersey: Prentice-Hall, 1972.
Pfeffer, J. Power in Organizations. Marshfield, Massachusetts: Pitman Pub. Inc., 1981.
Pfeffer, J. Managing with Power: Politics and Influence in Organisations. Boston, Massachusetts: Harvard Business School Press, 1992.
Sachs, P. "Transforming Work: Collaboration, Learning and Design." Communications of the ACM, Volume 38, Number 9, 1995, pp. 36-44.