Towards a Model of Requirements Elicitation for Software Procurement

G. Michael McGrath¹

¹) Victoria University, School of Information Systems (michael.mcgrath@vu.edu.au)

Abstract

Requirements elicitation for COTS is a much under-researched area. However, for software vendors the ability to undertake rapid requirements elicitation (RRE) in response to an RFT is absolutely critical. This appears to be an area where practice is well-advanced on research. A case study is used to demonstrate that gaining access to key knowledge sources in vital in RRE. Here, social network analysis techniques may be used to good effect in mapping relevant communications networks and in quantifying influence levels between key players in the procurement process. The communications network is also a major driver of the organizational decision making paradigm that underpins our software procurement model: specifically, a variant of the ‘garbage can’, derived ultimately from the original version presented 30 years previously, but intuitively appealing and still attracting much interest from contemporary researchers in a variety of fields.

The abstracted RRE model presented here is based on the episodic/encounter software procurement model specified by Heiskanen et al. [5]. In addition, by representing key elements of their model using a system dynamics formalism, the powerful simulation capabilities of modern system dynamics software packages can be taken used to advantage. Specifically, the ithink modelling and simulation package is employed. Motivation is a major determinant of the effort that vendors apply to managing relationships with their customers. Expectancy theory [3] is one of the better-known and more useful theories of motivation, it maps very neatly to a system dynamics representation [8], key aspects of the episodic/encounter model are quite consistent with expectancy theory and, consequently, it is utilised as a framework for our own procurement model. Finally, our model, while still a prototype and requiring considerable extension, has been developed to the stage where it may be employed to simulate important facets of the software procurement process and as a pedagogical aid in ‘games’ mode.

1. Introduction

As noted by Farbey and Finkelstein [4], most research into requirements elicitation (and software engineering in general) has focussed on the development of systems constructed in-house, custom-designed for a particular organization. Very little attention has been directed towards packaged solutions: despite the fact that these systems have been around for a very long time and with the proliferation of Enterprise Resource Planning (ERP) solutions through the 1990s, more and more companies seem to be turning to so-called ‘Commercial Off-the-Shelf Software’ (COTS) in preference to in-house development.

In one sense, packaged solutions should simplify the requirements elicitation process. With mature products particularly, vendors will generally have a very good understanding of both their product's functional scope (i.e. the requirements covered) and the translation of that functionality into its system design and physical implementation. In theory, the elicitation process then becomes a simple matter of identifying discrepancies between the client's needs and package functionality. Experience, however, indicates that this is far from trivial exercise (see e.g. [17]). To further complicate matters, client organizations are increasingly demanding very rapid responses to RFTs for package-based solutions and, moreover, expect submissions every bit as comprehensive as specifications prepared in-house for custom-built solutions (generally developed over a period of several months).

In these circumstances, it would appear that any software vendor hoping to win a package-based bid must have excellent access to key sources of relevant knowledge within the client organization. McGrath and More [10] report on a case study into the extent to which access to key knowledge sources determined outcomes in an ERP contract bidding exercise and where the bidders were given very little time to prepare their submissions. They termed this process rapid requirements elicitation (RRE) and here we present an RRE model which is, essentially, an abstraction derived from this case study and concepts extracted from a variety of software engineering and organization and management theory (OMT) literature bases.
Our paper is organized as follows: in Section 2, we present a summary of the case study referred to above. Then, in Section 3, we present some preliminary conclusions drawn from the study. In Section 4, a broad specification of our first pass at an RRE model for software procurement is presented and, finally, Section 5 contains concluding comments.

2. The Study: ERP Developments in Gigante Corporation

This case study summary is taken from [10]. The research in question was conducted as a retrospective and exploratory case study [19]. The broad research question addressed was:

To what extent does access to key knowledge sources determine outcomes in information systems contract bidding exercises based on rapid requirements elicitation?

The client in the case study was a large Australian IT company, which we shall call Gigante Corporation. A Partial view of Gigante’s organization structure is presented in Figure 1. The original study was concerned mainly with the IS operations of the international division, Gigante International (GInt). Six software firms were contenders for a multi-million dollar ERP contract to be let by Gigante with one bidder, Worldwide Information Technology (WIT), involving both its Sydney and Melbourne branches (to some extent, as competitors). Our principal source was the Manager of WIT’s Melbourne branch, and information was gathered from him and his colleagues in a series of six lengthy half-day workshop interviews conducted between March and July, 2001. Where possible, this input was corroborated (and added to) through interviews with other involved parties in WIT and the client company, Gigante. We were also given access to important background procedural documentation and other files produced as part of the bidding process.

Our sources within WIT and Gigante (five in total) were asked to estimate direct access levels from each vendor to all key parties within Gigante at the end of each of the four phases the study was divided into. The means of these estimates were then used as the basis of the adjacency matrices from which influence levels (and various other measures) were derived. Sources were also asked to estimate vendor’s levels of satisfaction with the outcomes of each phase and satisfaction and influence levels were then correlated with each other. Within limitations, this then provided a (rough) quantitative base to assist in addressing our research question.

2.1 Phase p1: 1996 – November 1999

The study covered a period from June 1996 through to end-2000. In June 1996, Gigante outsourced all its internal IS development and operations work to WIT. GInt's systems had all been converted to run on a specific ERP product base during the early 1990s and these (100+) systems, together with the 40 ERP specialists responsible for their development and maintenance, were transposed to WIT as part of the outsourcing deal. This group was established in Melbourne as a new, largely autonomous unit, within WIT and we shall refer to them as WIT-Melbourne (WITM). Importantly, WIT already had an ERP unit, based in Sydney, specialising in exactly the same product base as GInt's systems had been developed upon. We shall refer to this group as WIT-Sydney (WITS).

---

1 For a more complete account of our data analysis methods, the reader is referred to [10].
In the three years immediately following commencement of the outsourcing arrangement, a number of major IS initiatives were instigated by different groups within GInt:

1. A finance system, GlntFinS, was built for the organization by a consultancy firm we shall call Cons1. Originally implemented in April 1997, the system cost $4.5M to run and was only ever deployed in 4 out of 10 off-shore sites. This project was initiated, funded and controlled by a group within GInt independently of the WIT deal. When it turned out to be a dismal failure (as evidenced by the refusal of 6 intended sites to deploy it), WITM (with suitable recompense) agreed to pick up the pieces and maintain the system as part of their portfolio.

2. By the late 1990s, the ERP systems maintained by WITM were in urgent need of an upgrade to a new version of the vendor’s software. Under the original terms of the outsourcing agreement, WIT were supposed to have a monopoly on all Gigante ERP work for a period of 10 years. In late 1998, however, a major board reshuffle occurred. This included the appointment of a member of the consultancy firm, Cons1 and, in early 1999, the Board dictated to Gigante’s (IT) Steering Committee (GSC) that any major new IT project (value $1M) had to go out to tender. The consequential RFT for the software upgrade (in June 1999) resulted in a winning bid by a consortium comprised of: i) WITM; ii) a second ERP consultancy firm, Cons2; and iii) the ERP vendor’s consultancy arm, ERPC. With their expert knowledge of the new ERP software base, ERPC’s presence in the consortium made sense. WITM had no good reason to include Cons2 in their winning bid – apart from less than subtle pressure from Gigante!

3. With the failure of GlntFinS, GInt was in need of a much-improved financial system. In August 1999, Gigante released an RFT, with requirements based very much on GlntFinS, but substantially upgraded to take account of end-users’ major concerns. Bids were submitted by WITM, ERPC, Cons1, Cons2 and two additional ERP consultancy firms, Cons3 and Cons4. In November 1999, all bids were rejected because they were either too high or poorly scoped. The system, as specified in the RFT, has never been built.

At this point, Phase $p_1$ came to an end and Gigante realized it had something of a debacle on its hands. The legacy systems maintained by WITM were running smoothly enough but were in urgent need of an upgrade (yet to get underway). Legacy system functionality in the finance area overlapped substantially with replacement systems (one proposed and another already implemented, neither of which had effectively addressed any real IS need). Millions of dollars had been wasted - most of it as a direct result of decisions taken within GInt. This greatly strengthened the hand of the Gigante service departments, GFin and GSupp, with the IT Steering Committee, GSC. A sociogram depicting relevant access paths at the end of $p_1$ is presented in Figure 2. The reader may care to note the following features of the

![Figure 2: Access paths at end of p1.](image)

Our WITM source speculates that Gigante’s reasons probably included a desire to lessen their dependence on WIT and a strand of even-handedness that was intrinsic to the organization’s culture (discussed in more detail later).
diagram:

- The *Chameleon* (*Cmln*) was an ERP programmer employed by GInt before being transferred to WITM as part of the outsourcing deal. Based in Sydney, he spent most of his time on GInt premises – to the extent that most GInt staff thought he still worked for Gigante. This gave WITM excellent access to GInt.

- The head of WITM had previously worked for both Gigante’s Finance and Supply Departments (*GFin* and *GSupp*). He had maintained very good contacts and information sources in both these areas.

- Although they had submitted bids for the software upgrade contract, neither Cons3 nor Cons4 were serious players within the case study domain.

- *WITE* were a group of *engagement specialists*, expert in managing access to key knowledge sources during bid preparation. WITS were experienced in making good use of these specialists but neither group had played any significant part during phase $p_1$.

2.2 Phase $p_2$: December 1999 – end-January 2000

As noted above, a strong thread of probity, even-handedness and fairness was deeply embedded in Gigante’s culture - nowhere more so than in *GFin* and *GSupp*. As such, they were less than enthusiastic about the exclusive nature of the WIT outsourcing deal. To address GInt's difficulties, the two departments proposed that a *blueprint* be prepared and that this should serve as the basis for a further RFT. The blueprint effectively was to be a document covering all of GInt's ERP needs and, in an attempt to avoid the scoping problems that had helped scuttle all bids for the upgraded financial system in the previous phase, Gigante requested that each of WITS, Cons1 and Cons2, provide two ERP experts to participate in a working party commissioned to produce the blueprint. All three companies accepted the offer and produced the document in the allotted six weeks.

The sociogram at the end of $p_2$ is presented in Figure 3. The following key points should be noted:

- The establishment of a *Tender Evaluation Team* (*TET*), reporting to the IT Steering Committee and comprised of GInt, GFin and GSopp members.

- The Chameleon having access to the TET because of his close links with GInt.

- The involvement of WITS and WITE, their links into GInt (for the first time) and WITS’s access to WITM. Effectively, from this point, WITS had become the major (official) WIT player in the bidding process. To a large extent, this was due to GInt’s preference for dealing with Sydney-based organizations.

2.3 Phase $p_3$: February 2000 – end-June 2000

![Figure 3: Access paths at end of p2.](image-url)
By February 2000, the blueprint had been completed and the RFT had gone out. All the “usual suspects” (i.e. the six firms that tendered for the enhanced finance system in p2) prepared to submit bids. From a software engineering viewpoint, the most interesting aspect of this phase was that the bidders were given only two weeks to produce an extremely comprehensive document encompassing: detailed functional and non-functional requirements, a technical specification, a project plan, a staffing plan, the development methodology and configuration approach, timing and cost estimates and risk factors. This is the RRE process alluded to earlier and, to be successful, our contention is that an organization must not only be superbly organized but have very good access to key knowledge sources within the client organization – particularly to sources that have knowledge critical to the bidding process but, for a variety of reasons

![Figure 4: Access paths at end of p3.](image)

(WITM had no experience in preparing these bids. Thus, from WIT's perspective, it was probably fortunate that WITS were given the lead role. On the other hand, WITM had most of the functional knowledge of the application domain, very good (relevant) technical knowledge and excellent access to key knowledge sources in GInt, GFin and GSupp. WITS made maximum use of their Melbourne-based colleagues to the extent that, by the time the bid had been prepared, WITM (to use the words of their manager) had effectively “been sucked dry of their most critical knowledge”. In addition, through their ‘engagement experts’, WITS used the bid preparation process to strengthen and consolidate their fledgling relationship with GInt.

All contractors completed and submitted their bids by the mid-February deadline. The irony was that Gigante's tender evaluation team then took the better part of three months to assess the bids - and then announced that, because all bids were too high, none had been successful! Then, in what appears to be an extraordinary move, GSupp invited Cons2 (alone) to rebid. Not surprisingly, all other bidders "screamed" and Gigante was forced to invite all parties to rebid. After reflection, two initial bidders, Cons3 and Cons4, decided not to proceed. A further side-effect of this mini-debacle was that GSupp suffered a major blow to their credibility and, thereafter, played no significant part in events. Our sociogram at the end of phase p3 had now evolved to the form presented in Figure 4.

2.4 Phase p4: July 2000 – October 2000

Possibly because of embarrassment suffered during the previous phase, Gigante were a good deal less demanding during the rebidding process (at least, in terms of timeframes). Specifically, the three remaining bidders (WITS, Cons1 and Cons2) were given two months to prepare their revised bids. During this period, a chance event occurred that, in our view, influenced the ultimate outcome more than any other single factor.

3 Their rationale was that Cons2 was already working in the organisation. This, however, completely ignored the fact that ERPC and WIT (through WITM) were in the same position and that the latter, in particular, had much superior knowledge of the application domain.
We have already talked about the Chameleon (Cmln): an ERP specialist, transferred from GInt to WITM as part of the outsourcing deal, but still regarded by GInt as "one of theirs". During a casual conservation with the Chameleon, a GInt executive revealed the fact that the upper limit of their ERP project work budget was $6.6M. The Chameleon immediately passed this on to his manager, the head of WITM who, in turn, passed it on to the WITS bid team. WITS reduced the scope of their proposal, submitted a bid for $5.9M worth of work and were duly awarded the contract in October 2000. Thus, at this point, the only relevant portion of the total communications network was that presented in Figure 5.

While WITM staff were disappointed to some extent that their 'rivals' in Sydney had taken over their work, senior executives in the company were delighted with the efforts of all involved. Consequently, the head of WITM was handsomely rewarded and assigned to an equally-responsible managerial position and all his programmers were either redeployed within other areas of WIT or found work in other companies. In December 2000, other events inside Gigante conspired to disrupt this happy outcome for WIT but this is outside the scope of this paper (and largely irrelevant to the original case study).

3. Preliminary Conclusions

The above is a much-abbreviated version of the original case study narrative. However, it does appear that software companies, in response to clients' demands, do indeed seem to applying a new form of very-rapid requirements elicitation. The documents produced by WIT during the bidding process amount to comprehensive and detailed specifications covering functional, technical and management requirements. In addition, the processes employed to produce these specifications were impressive; in terms of both their thoroughness and the speed with which they were put together. Moreover, since the quality of WIT's proposal was by no means the major determinant of the ultimate RFT outcome, we have no reason to assume that WIT's competitors are any less advanced in this form of RRE.

Finally, returning now to the case study research question, the study did seem to indicate that social network analysis techniques may indeed be used to good effect to study communications links during requirements elicitation. The study focus was on vendors' access to key knowledge sources within a client organization and the series of sociograms developed through the five phases of the study clearly illustrated the evolution of the essential communications network through the study period. Importantly, the influence algorithm [6] proved particularly useful in calculating overall access levels and this specific approach is probably the most appropriate (of the many available alternatives) for the particular type of communication network studied in this instance. In addition, Table 1 below, reproduced from [10], clearly indicates an increasingly close correlation between levels of influence and satisfaction through the study period.

Our findings, however, are limited in their internal and external validity. Specifically: i) the study was retrospective, which makes us somewhat too reliant on our sources' memories; ii) findings should not be generalised beyond ERP applications; and iii) all data collected, plus estimates of access and satisfaction levels, was gathered from sources within WITM and Gigante. Ideally, we should have had similar input from all involved parties. Despite these limitations though, our case study does seem to suggest that access to knowledge sources is a key critical success factor in rapid requirements elicitation and, perhaps more importantly, indicates several additional promising areas for further investigation. In the following section, we focus on one of these.

Figure 5: Relevant access paths during p4.
Table 1: Influence-Satisfaction level correlations.

<table>
<thead>
<tr>
<th></th>
<th>IP2</th>
<th>S2</th>
<th>IP3</th>
<th>S3</th>
<th>IP4</th>
<th>S4</th>
<th>IP5</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>WITS</td>
<td>0.077</td>
<td>0.8</td>
<td>0.194</td>
<td>0.4</td>
<td>0.236</td>
<td>1</td>
<td>0.333</td>
<td>1</td>
</tr>
<tr>
<td>WITM</td>
<td>0.332</td>
<td>0.5</td>
<td>0.199</td>
<td>0.2</td>
<td>0.164</td>
<td>0.6</td>
<td>0.329</td>
<td>0.7</td>
</tr>
<tr>
<td>ERPC</td>
<td>0.166</td>
<td>0.4</td>
<td>0</td>
<td>0.3</td>
<td>0.1</td>
<td>0.2</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>Cons1</td>
<td>0.077</td>
<td>0.8</td>
<td>0.059</td>
<td>0.2</td>
<td>0.071</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cons2</td>
<td>0.083</td>
<td>0.8</td>
<td>0.062</td>
<td>0.4</td>
<td>0.076</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cons3</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cons4</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$r$ values: 0.1764, 0.4326, 0.9068, 0.9628

**Keys**
- $IP_n$: Influence level during Phase $n$
- $S_n$: Level of satisfaction with Phase $n$ outcome

4. **Towards an ‘RRE for Software Procurement Model’**

As noted earlier, requirements elicitation for COTS is a much under-researched area and, we speculate, an area where practice is well advanced on research. Nevertheless, there are indications that important research questions within the area are at last receiving the attention they deserve. For example: Farbey and Finkelstein [4] propose that if a COTS implementation is to be ‘world class’, software, software development processes and the software supply chain should all be designed together; the question of how best to integrate requirements elicitation with business process modelling and analysis has recently generated much debate in the ERP world [12]; and, again concerning ERP applications, Parr and Shanks [14] have emphasised the need to tailor requirements elicitation (and, indeed, the entire development approach) to application scope (ranging from comprehensive and cross-organizational to very limited).

While it is still early days, some common themes have begun to emerge from this limited literature base. These include:

- COTS vendors are generally subject to very tight deadlines (see e.g. [7]), making RRE a necessary survival skill;
- Management of relationships with key personnel in client organizations is critical [4, 7];
- Vendor-client relationships may gradually deteriorate unless carefully nurtured [5] and this may inhibit access to key client personnel at (re)bid time; and
- Chance and random events are significant determinants of outcomes during all phases of COTS customisation, implementation and operation [12].

All this is quite consistent with the preliminary findings of our case study research presented earlier: specifically, organizational (and inter-organizational communications networks) are absolutely critical.

In a particularly interesting piece of work focussed on COTS, Heiskanen et al. [5] present a ‘social model of software development’ based on user-developer interaction. This interaction is defined as relatively stable “state progress passages” over periods of time (episodes) punctuated by “critical events” (encounters). Encounters have the potential to radically change various attributes of user-developer interaction; such as developer access to users, influence and user satisfaction with developer performance.

Longitudinal representations of interactions are portrayed in diagrams similar to Figure 6. In this example, the user organization is initially equivocal about its new system. Then, at encounter No.1, something happens (e.g. a major disagreement over project direction) that causes users to reject the system. Later, at encounter No.2, rejection changes to acceptance (because e.g. the initial version is released and users are very happy with it). At encounter No.3, the user organization again rejects the system (due e.g. to a sharp increase in leasing and maintenance fees).

These diagrams are very useful but key features of the episodic/encounter model seem to be eminently well-suited to a system dynamics representation. In particular, the ability of system dynamics modelling tools capture the complex feedback loops intrinsic to software processes and to automatically simulate system behavior over time may be used to good effect. In addition, more subtle aspects of the vendor-client relationship can be neatly modelled, simulated and illustrated. Here, there is already a substantial and rapidly-expanding body of work dealing with the use of system dynamics for (in-house) software engineering process modelling; the seminal work in this field being that of Abdel-Hamid and Madnick [1].
Our initial version of a RRE model for software procurement is illustrated in Figure 7. It was developed using the system dynamics modelling tool, *ithink*. The basic building blocks of system dynamics models are stocks (represented as rectangles), flows (represented as arrows with circular flow regulators attached) and converters (represented as circles). The stocks in our model are *Work*, *LOA* (level of access), *Effort* and *Reln Mgnt Cost* (relationship management cost). There is a level associated with each stock, which can be an actual value (e.g. *Effort* is the cumulative number of hours the vendor puts into managing the client relationship) or a value bounded by some artificial scale (e.g. *LOA*, the vendor’s level of access to the client, is measured on a 0-1 scale, with 0 meaning no access and 1 meaning excellent access). Stock levels vary with flows, which may be inflows, outflows or bi-directional. For example, *LOAVarn* is a bidirectional flow such that:

\[
LOA_t = LOA_{t-1} + LOAVarn_t
\]

These equations are the foundation of *ithink*’s formidable simulation capabilities. The third of our basic constructs, converters, serve a utilitarian role: they hold values for constants, calculate mathematical relationships and serve as repositories for graphical functions. In general, they convert inputs into outputs (hence, the name, “converter”). For a detailed introduction to system dynamics modelling, the reader is referred to [11].

Our model is underpinned by two theories from the OMT literature; expectancy theory and the ‘garbage can’ model of organizational decision making. Expectancy theory [3] is actually a group of motivation theories founded on a rational-economic view of people but all versions are based on: i) the expectation that an act will result in a given outcome; and ii) the attractiveness of that outcome to the party. Effort is linked to performance and performance, in turn, is linked to rewards. If a breakdown occurs at any point in the chain then, ultimately, less effort will be put in (i.e. motivation will suffer). These linkages and loops are clearly evident in our model presented in Figure 7. Specifically: i) a vendor puts effort into managing a client relationship (which costs money); ii) the effort expended is a major determinant of the level of access that the vendor has to the client (performance); and iii) the level of access, in turn, will have an impact on the amount of work the vendorsecures from the client (rewards). If, however, the vendor puts a lot of effort into the the relationship and this does not bear fruit (in both improved access to the client and more work), then he or she is liable to be demotivated. Finally, in our base model, the user (playing the role of the vendor) has the opportunity to vary the effort put into maintaining the relationship at different points during a simulation run. The temptation to do this may be considerable because relationship management is expensive and consequences (reduced level of access and less work) may not become evident for some time.

The parameters of our base model have been established such that vendor effort will, indeed, deteriorate gradually over time – behaviour quite consistent with the poor customer relationship management characteristic of much COTS implementation and maintenance [5]. Obviously, if this continues for long enough without major intervention (or encounters, using the terminology of Heiskanen et al.), then the vendor is liable to be at a major disadvantage if (for example) the opportunity to bid for a major new contract arises.

We have designed our model so that encounters may be conveniently introduced at any point during a simulation. When this occurs, the vendor (user) must respond (and may also vary relationship management effort at the same time). For example, using material from our case study, we have developed a version of our model where the vendor must deal with the following situations:

---

4 This behaviour is built into the feedback loops from *Work* and *LOA* to *Effort*. 
1. A highly-disruptive system crash occurs. The vendor must balance the (not inconsiderable) cost of fixing the problem quickly (by utilising significant numbers of the company’s best and brightest software engineers) against the longer-term impact on the LOA of a failure to do this.

2. Responding to the “even-handedness” decision discussed in 2.2. Here, the vendor’s LOA will increase if the decision is accepted with relatively good grace. Alternatively, resisting the decision too strongly will ensure the LOA decreases.

3. A major RFT is released (see 2.3). The vendor’s LOA at this point will largely determine the bid outcome.

Thus, our model may be employed as a management game in which the user plays the part of a software package vendor, dealing with one specific client in a competitive environment. The user has control over the amount of effort put into relationship management and has to respond to each of the situations detailed above. Each response will have an impact on the organizational communications network and, specifically, will strengthen or weaken the vendor-client link (as designated by the LOA). The game climax comes with the RFT: lose the bid and you lose the customer. Success can be measured in a variety of (related) ways, but the principal indicator of success is:

\[ \text{Net Profit} = \text{Work} - \text{Reln Mgmt Cost}. \]

While expectancy theory, complemented by the episodic/encounter model of Heiskanen et al. [5], provides the framework for our model, system behaviour is actually driven by our variant of the ‘garbage can’ model of organizational decision making, first specified by Cohen, March and Olsen (CMO) 30 years ago [2].

CMO note that much decision making does not conform to a neat logical view, founded as it is in economics theory. Thus, over the years our view of the rational organization has been gradually eroded. However, as Warglien and Masuch [18] have noted, it is important to distinguish between organizational anarchy and organized anarchy. The former term is used to describe an organization in chaos while the latter term recognises that, despite the seeming irrationality and confusion present in many organizations, these are often underpinned by order and intelligence - or, as they describe it, an alternative “underlying behavioral logic” [2; p.3]. CMO’s garbage can model represents one of the more influential attempts to define such an alternative decision making logic. They view:

\[ \ldots \text{a choice opportunity as a garbage can into which various kinds of problems and solutions are dumped by participants as} \]
they are generated. The mix of garbage in a single can depends on the mix of cans available, on the labels attached to the alternative cans, on what garbage is currently being produced, and on the speed with which garbage is collected and removed from the scene [2; p.2].

Thus, from this perspective, an organization can be viewed as largely independent streams of choice opportunities looking for problems, problems looking for decision situations, solutions looking for problems to which they might attach themselves and decision makers looking for (or avoiding) work. Decisions, made when elements of all four streams come together, are of three styles: i) resolution, where a problem is worked through until it is solved (the style which most closely matches rational choice decision making methods); ii) oversight, where decisions that don’t really address any problem are made; and iii) flight, where persistent, unsolved problems move from one decision making arena (garbage can) to a new, more attractive choice opportunity (another garbage can). Solutions are only really effective when the first of these styles (resolution) is employed.

CMO implemented their model as a Fortran computer program and used it to simulate and analyse some interesting properties of emergent decision making processes [2]. Their work aroused a great deal of interest and a number of extensions to the garbage can were modelled and implemented through the remainder of the 1970s and 1980s. Warglien and Masuch [18; pp.18-23] present a summary of the more significant of these developments and note that all models share a decision making strategy based principally on numerical algorithms. In order to extend the simulation capabilities of the garbage can and to overcome problems experienced with numerically-based models (including a lack of model clarity and transparency, and the application of many simplifying assumptions), Masuch and LaPotin [13] adopted a modelling approach based on artificial intelligence (AI) techniques. Their implementation represents a significant advance, as it allows the explicit (and largely) declarative representation of organization structures, issues and problems, actors attributes and feasible decision alternatives.

Recently we specified further extensions to the garbage can model in an effort to improve its utility as an analysis tool and as a decision making and pedagogical aid [9]. We focussed particularly on problem content and action alternatives, and did so within a power-political framework. We employed much of the original CMO model and our use of AI techniques is very much in the tradition of Masuch and LaPotin [13]. We also relied heavily on the particular interpretation and representation of organizational power presented by Pfeffer [15, 16]. Our choice of model focus was informed by what we perceived to be a lack of any real “meat” in the problems and decision alternatives used to explicate previous representations of the garbage can and its variants.

The model presented here may be viewed as an extension of our earlier work. Technically, we continue to rely on deductive, rule-based knowledge representation methods but we now represent these (sub)-models within a system dynamics framework – effectively allowing us to take advantage of both of the commonly-used paradigms employed in simulating organizational behaviour; the numerical and symbolic approaches [18]. Perhaps more importantly though, we have extended our focus from in-house, custom-built systems to COTS. Our justification for this adjustment to our research direction was presented earlier in the paper.

5. Summary

A software process model is of limited use if it merely takes a snapshot of the process being studied. A longitudinal and dynamic representation scheme is preferable. Thus, the abstracted RRE model presented here is based on the episodic/encounter software procurement model specified by Heiskanen et al. [5]. By representing key elements of their model using a system dynamics formalism, the powerful simulation capabilities of modern system dynamics software packages can be taken used to advantage. Expectancy theory maps very neatly to a system dynamics representation [8], key aspects of the episodic/encounter model are quite consistent with expectancy theory and, consequently, it is utilised as a framework for our procurement model. Our model may be employed to simulate the software procurement process, with a focus on the vendor-client relationship and as a pedagogical aid. Using such models can clarify perceptual differences and behavioural concerns, and provide another useful aid in the manager’s toolkit of sanity producing aids necessary in the increasingly dynamic, complex, uncertain and daunting world of software engineering.

References


