Working paper in Borås Studies of Information Systems, No 20,
University College of Borås, Sweden

Legacy Systems: Assets or Liabilities?

A Language Action Perspective on Respecting and Reflecting Negotiated Business Relationships in Information Systems

Anthony Lauder

Computing Laboratory
University of Kent at Canterbury
Canterbury, Kent, CT2 7NF, UK
Anthony@Lauder.u-net.com

Mikael Lind

University of Borås
School of Business and Informatics
S-501 90 Borås, Sweden
Mikael.Lind@hb.se

Abstract

Business processes emerge to negotiate and fulfill contractual commitments underpinning developing business relationships. These business processes are supported by information systems. In *legacy* information systems the underlying transactions of contract negotiation and fulfillment are typically submerged in the accidental architecture of the information system itself. Furthermore, this accidental architecture impedes future support for inevitable business process change. In this paper we apply a Language Action Perspective to the issue of legacy systems and propose a series of participatory workshops wherein these lost transactions are reconstructed, refactored, and taken as the starting point for the negotiation of requirements for a replacement system with a purposeful architecture. The outcome of this approach is an ongoing process of reflection upon business change and its impact upon the organization, and a shared commitment amongst stakeholders on how to manage that change and reflect it in both the organization's business processes and in the information systems which support them. This paper is the result of a joint collaboration between the authors during a two-day meeting in the UK in late 1998, followed by a four-day meeting in Sweden in the spring of 1999.

1. Introduction

Computerized information systems are developed to support an organization's business processes. As business processes evolve, the information systems supporting them must evolve accordingly. Unfortunately, information systems tend to grow old "*dis*gracefully", so that over time it becomes harder and harder to modify them to reflect on-going business process change. Such systems are termed *legacy information systems* [Brodie and Stonebraker, 1995]. Businesses often describe their legacy systems as business assets. These same systems are also often referred to as business liabilities, sometimes even in the same sentence: "that system is our biggest business asset, and it's getting to be a real liability".

We found this use of terminology interesting and decided to probe deeper. What we discovered is that legacy systems are, indeed, simultaneously business assets and business liabilities. This triggered us to further investigate the nature of legacy systems and their replacement with contemporary information systems within the context of their satisfactory support for business processes undergoing continual change.

One of the authors has been working for some time with a mid-sized software development organization that has gained a major share of the market for a particular type of commercial information system. This organization tailors its information systems in collaboration with business clients who use and adapt those information systems to support their own business processes. Building upon its success, the organization has merged with a number of its competitors and hence secured an even larger market share. This organization, though, is faced with the problem that it now owns a number of information systems which have considerable overlap in their functionality, but which are highly divergent in their implementations of that functionality. These information systems are written in a variety of programming languages on divergent operating systems and export incompatible interfaces. Thus, the information systems cannot interact with one another. Furthermore, these information systems have each been developed over many years and are now perceived as legacy systems - each is viewed as "creaking under its own weight", to use a term borrowed from one of the developers involved. In particular, a major problem with these legacy systems is that it becomes increasingly more and more difficult to modify them to reflect each client's ongoing business process change. Supporting all of these applications is a significant drain on the resources of the organization; development staff is stretched thinly over the current maintenance burden, with few resources remaining for new development work. The legacy system dilemma faced by this organization, and consequently impacting its business clients, was the original inspiration for our decision to look for a fresh perspective on the legacy system issue.

Organizations are in fear of their legacy systems. They are afraid of keeping them, since maintaining them is a significant drain on the organization's resources. They are also afraid of replacing them. We will argue in this paper that a major reason organizations fear the replacement of legacy information systems - despite the resistance of such systems to reflecting on-going business process change - is that those legacy systems have undergone years of debugging effort. Through this debugging effort they have grown to support not just explicitly understood business processes but also *tacit business knowledge*, the origins and conscious awareness of which are lost in the history of the organization. It is this hard-earned reflection of tacit business knowledge that makes legacy systems such a critical issue for an organization.

Legacy systems, then, are not an issue that will simply go away; they are a very real problem facing many organizations and, thus, we need to develop an appropriate strategy for dealing with them. To develop such a strategy we need to adopt a philosophical stance from which that strategy will be derived. In this paper we adopt a philosophical framework based upon the Language Action Perspective [Goldkuhl et al. 1998c], with which we are able to gain an understanding of the business processes underlying an organization and supported by that organization’s information systems. It is via this philosophical framework, and the resultant understanding of and reflection upon an organization and its information systems, that we are able to manage the legacy system issue.

The paper begins, then, by adopting a Language Action Perspective (LAP) on organizations and their supporting information systems. Underpinning this adopted perspective is a view of organizational business processes as following a cooperative transaction pattern drawn from Business Action Theory (BAT). The paper argues that information systems that resist reflection of ongoing business process change are termed legacy systems. Next, it is argued that legacy systems form a business liability, which a business would ideally like to eliminate, but this elimination proves to be a major challenge without simultaneously losing the reflection of tacit knowledge which makes the legacy system a business asset. Consequently, the paper proceeds to propose a series of BAT-transaction-oriented participatory workshops via which the tacit knowledge reflected in a legacy system may be captured, reflected upon, refactored to accommodate current and anticipated future business process change, and supported by appropriate information systems. The paper concludes with reflections upon the consequences of the participatory-workshop-oriented approach, and a preview of further research that follows from the adoption of this approach.

2. Understanding organizations and information systems

2.1 The language action perspective

As a philosophical basis for understanding organizations and their supporting information systems, we are drawn to the Language Action Perspective (LAP) [Goldkuhl et al. 1998c]. LAP has its roots in many works, including Speech Act Theory [Austin, 1962; Searle, 1969], the Language-Game theory of Wittgenstein [Wittgenstein, 1958], the Theory of Communicative Action of Habermas [Habermas, 1984], and the work of Winograd and Flores [Winograd and Flores, 1987].

LAP focuses on communicative action as the key concept for the understanding and modeling of organizations and their supporting information systems. This requires a theory explaining communication and its functions. The Speech Act Theory [Austin, 1962; Searle, 1969; Searle, 1979] has proven to be a compelling frame of reference for this purpose (e.g., [Flores and Ludlow, 1981])[Winograd and Flores, 1987; Winograd, 1988]. The primary characteristic of Speech Act Theory is that it considers language use to be a form of rule-governed behavior. Uttering a sentence is seen as the performance of a purposeful act, a so-called speech act. The important characteristic of a speech act is recognition of the fact that speech is purposeful. That is, every utterance is made with an underlying intention to induce an appropriate responsive action. The initial impetus for a speech-act-based conceptualization of organizations has been the work by Flores and Ludlow [Flores and Ludlow, 1981] which perceives organizations as networks of inter-related commitments established and fulfilled by various speech acts (directives, commissives, assertives and declaratives) and supporting material acts.

LAP considers the utterance of speech acts to be the fundamental backbone of business processes and, consequently, LAP-based modeling of business processes places strong emphasis upon speech act use. There are many branches of LAP, including Action Workflow [Medina-Mora et al. 1992], Business Action Theory [Goldkuhl, 1996; Goldkuhl, 1998a], and DEMO [Dietz, 1994; van Reijswoud and van der Rijst, 1999]. All of these branches are bound by the underlying idea that actors use speech acts with purposeful intent to communicate needs, align goals, coordinate activities to satisfy those goals, and build enduring personal relationships based upon mutual trust. This final point – the development of personal relationships based upon mutual trust – is a particular emphasis of Business Action Theory. We consider this an important strength of

Business Action Theory, and believe that it holds particular promise for the resolution of the legacy system issue. Hence, we adopt Business Action Theory as our underlying philosophical framework throughout this paper.

2.2 Exchange Relationships from a Business Action Theory Perspective

Business Action Theory (BAT) [Goldkuhl, 1996; Goldkuhl, 1998a] argues that businesses use language to play a conventionalized *action game* (a universally understood, almost ritualistic, pattern of behavior with multiple actors sharing and adhering to mutually recognized and accepted rules). That action game encompasses the pursuit, negotiation, and fulfillment of exchange agreements. The underlying philosophy is that organizations cooperate to arrive at mutual success and to establish long-term business relationships built upon mutual trust.

BAT is intended to describe and explain business cooperation (interaction) as well as forming a theoretical lens through which to view organizational change during business process development. BAT forms a useful interpretative framework when reconstructing, evaluating and redesigning business processes. In such situations, BAT must be supplemented by congruent change methods [Lind and Goldkuhl, 1997]. The perspective underlying BAT, when applied in this way, stimulates the formation of generative questions, thus directing our attention and our creativity towards the essential aspects (with a strong emphasis on communicative actions) underlying an organization and its business processes.

BAT perceives business interactions as pursuing a course described by the following multi-phase transaction framework or pattern:

1. *Business prerequisites phase* – where actors identify their lacks and their excesses
2. *Exposure and contact search phase* – where actors advertise their lacks as needs and their excesses as capacities
3. *Contact establishment and proposal phase* – where actors whose needs match one another's capacities negotiate a rate and schedule of exchange
4. *Contractual phase* – where actors commit to a mutual exchange agreement resulting from negotiation
5. *Fulfillment phase* – where the actual exchanges of value occur as agreed between the actors
6. *Completion phase* – where both actors are satisfied that the exchanges have occurred as agreed, or if they have not occurred the actors take some appropriate remedial action

Note that BAT emphasizes an ordering between different groups of communicative actions within a business process. The different phases of the transaction pattern constitute such groupings. The six phases of the BAT transaction pattern imply an on-going interaction between the actors, wherein there is a continual process of exchange between them. This process involves the actors exchanging knowledge, interests, proposals, counter-proposals, commitments, and so on, but the eventual aim (the aim of the whole transaction) is a mutually satisfactory exchange of value (perhaps money for products, but possibly something less tangible like goodwill). The exchange process is co-ordinated through the BAT transaction pattern. Each step (communicative action) in that pattern (e.g. offer, order, delivery promise and commitment) will always need to be performed in principal, but in simple business situations some of these actions may be implicit or integrated with other actions.

In the BAT transaction pattern, the process of negotiation between actors is aimed at determining a mutually acceptable exchange rate – aligning their perception of the relative values of their respective excesses – and a mutually acceptable schedule for exchange. The mutually satisfactory fulfillment of a negotiated exchange establishes trust and thus strengthens the exchange relation (the business relationship) between the actors. Underpinning that strengthened exchange relation is the history of transactions between the actors, including the history of negotiated rates of value exchange and negotiated sequences of exchange, forming customary practice between the parties. It is these historically constructed customary practices that are the basis of on-going business relationships. There is compelling evidence that businesses view negotiated agreements not as legal contracts (even in cases where they are clearly legally enforceable) but rather as "gentleman's agreements" dependent more upon a tacit code of honor than any recourse to law [Macaulay, 1992]. There is, it appears, an underlying recognition that it is the "human touch" and the establishment of long-term personal relationships, rather than red tape, contracts, and lawyers, that oils the wheels of industry [Macaulay, 1992].

The original intent of BAT is to understand commercial interaction in supplier/customer relationships in terms of contractual negotiation, commitment, and fulfillment. Note that the fulfillment of commitments is judged in terms of the customer's payment for services rendered by the supplier. It is possible, however, to liberate BAT from a purely supplier/customer interaction. The idea that two actors (participating in two roles) negotiate, commit to, and fulfil contracts is, it turns out, more generally applicable to any mutual collaboration involving an exchange of value. It is precisely this more general application that we appeal to in this paper.

The term *transaction* (as in the BAT transaction pattern) is used in this paper to encompass both contract negotiation and contract fulfillment. Actors negotiate and reach mutual contractual commitment to achieve a certain goal if certain contractual obligations are met in advance. Each actor has an initial obligation that must be met before a transaction can begin to fulfill a contract - these initial obligations form the *pre-condition* of the transaction. Each actor also has a final obligation that must be met for the transaction to be complete - these final obligations are the goal of the contract and form the *post-condition* of the transaction. Satisfaction of a transaction's pre-condition is recognition that a contract *can* be fulfilled. Satisfaction of a transaction's post-condition is recognition that a contract *has* been fulfilled. This view is consistent with the notion of *conditions of satisfaction* [Winograd and Flores, 1987] - the specification of which forms a early phase in a complete transaction – which, when met, indicates the successful fulfillment of a transaction. The body of a transaction, then, is a cooperative act between the actors involved which transforms the pre-condition into the post-condition.

2.3 Building Mutual Trust through Repeated BAT Transaction Success

A number of recent theories on organizational behavior, such as the concepts of living [Morgan, 1996] and learning [Senge, 1990] organizations, recognize that businesses are constantly adapting themselves to satisfy the ever-changing needs of their stakeholders. Our own work, both in academia and in industry, has led us to view organizations as *pursuing the fulfillment of contracts negotiated with stakeholders*. The process of negotiation between an organization and its stakeholders is never ending. As circumstances change, an organization adapts itself by negotiating new contracts. Both the organization and its stakeholders are continually monitoring their positions to see whether or not their current contracts offer the best value, and where this is not the case they will often either negotiate new contracts to achieve better value or take their business elsewhere.

The term *organization* is used here to denote a collection of mutually recognized actors bound by, and collaborating in fulfillment of, common long-term goals. The term *contract* is used here not in the strict sense of a legally enforceable commitment, but in a fairly broad sense to mean any agreement resulting from a negotiation. The term *stakeholder* is used in a fairly broad sense to mean any actor interested in, affected by, and possibly participating in, the activities of the organization - for example customers, suppliers, regulatory bodies, and employees of the organization itself. Stakeholders are the role-assuming actors in the BAT transaction pattern with whom the organization collaborates in pursuit of the organization’s goals. The term *negotiation* is used in a fairly broad sense to mean a process whereby two or more parties agree to a mutually acceptable exchange of value. Negotiation may be either explicit or implicit. In the case of explicit negotiation, for example, a customer may offer to buy goods at a certain price if the supplier meets certain conditions. In the case of tacit or implicit negotiation, diners simply stop eating at a restaurant where quality is low, and may be convinced to return when the restaurant announces a new chef.

Since an organization’s primary mission is to fulfil its long-term goals, an organization needs to nurture, respect, and strengthen the business relationships that facilitate the fulfillment of those goals. Such relationships are underpinned by continual application of the BAT transaction pattern (including negotiation and fulfillment of contracts) when collaborating with stakeholders. It is essential that, if the collaboration is to succeed, the shared focus be upon mutual satisfaction, rather than self-interest. It is this process of mutually satisfactory negotiation and fulfillment that encourages the establishment of mutual trust and thus strengthens (rather than weakens) business relationships, thus increasing the possibility of future mutually profitable collaborations between the organization and its stakeholders.

Each BAT transaction establishes and fulfills an immediate exchange contract. However, as Figure 1 shows, a business relationship between an organization and a stakeholder is continuously developed by an on-going succession of individual BAT transactions. Business relationships form mutual expectations of the relative positions of the participants in those BAT transactions. An individual BAT transaction is influenced by the business relationship established by preceding BAT transactions, while simultaneously strengthening or weakening that business relationship thus influencing subsequent BAT transactions.

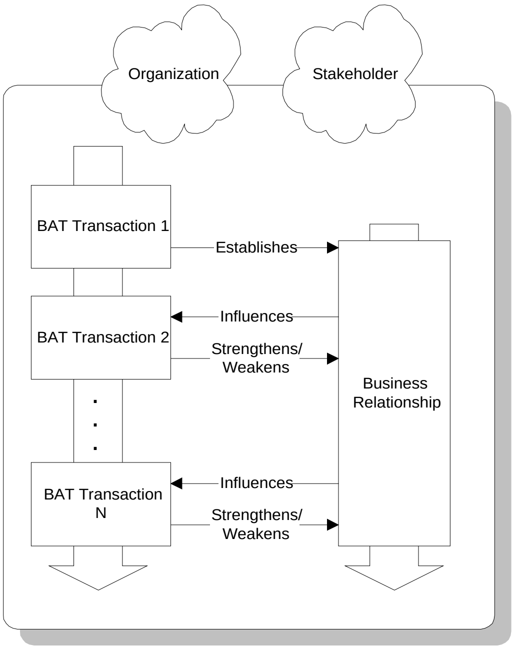

Figure 1: Business Relationship Development

2.4 A Language Action Perspective on Information Systems

An information system viewed from a Language Action Perspective [Goldkuhl and Lyytinen, 1982] is seen as an *information agent* [Verharen, 1997] which performs communicative acts contributing to the creation of value for the organization [Goldkuhl and Ågerfalk, 1998b]. Business processes are seen as being undertaken by human actors in collaboration with information agents (information systems). Thus, information systems can be viewed both as tools which support human actors in their collaborative acts, and also as agents which perform (rule-governed) actions independently of direct human involvement [Goldkuhl and Ågerfalk, 1998b].

By adopting a BAT perspective on business processes, we come to view an information system as a tool that supports an organization in the transactional negotiation and fulfillment of contracts with its stakeholders. Indeed, Goldkuhl and Agerfalk [Goldkuhl and Ågerfalk, 1998b] observe that an information system can be used in business interactions corresponding to all phases of the BAT transaction pattern.

The collaborative acts supported by an information system are the interactions between an organization and its stakeholders according to the BAT transaction pattern. Since individual collaborative acts also contribute to and are influenced by developing business relationships, an information system must not support just individual collaborative acts, but must also respect the influence spanning across individual BAT transactions inherent in those continually evolving business relationships.

The total communicative actions (collaborative acts) in which an information system has the ability to participate forms its overall *action repertoire*. This action repertoire determines the BAT transactions in which an information system may participate. Furthermore, an information system must maintain an *action memory* capturing the impact of its historical collaborations [Goldkuhl and Ågerfalk, 1998b]. The current state of an information system's action memory determines the collaborations it is willing to undertake next. The actions which an information system is able to participate in at a given moment, as governed by its action memory, form its current *action potential*. For example, if a collaboration resulted in a supplier reaching a sale agreement with a customer, then the action memory of that agreement forms the basis for the expectation of the actual fulfillment of that sale. Thus, a historically determined action memory dictates the flow of the interaction protocols (following the BAT transaction pattern) in which an information system participates. An information system must maintain both a short-term and a long-term action memory. A short-term action memory reflects the progress of an individual BAT transaction, whereas a long-term action memory spans across successive BAT transactions and thus reflects a developing business relationship. The long-term action memory captures the commitments, fulfillments, and future expectations resulting from a succession of individual BAT transactions. It is this long-term action memory (i.e. the reflection of business relationships) which influences individual BAT transactions which in turn strengthen or weaken the business relationships reflected in that action memory.

The action potential of an information system also reflects these two positions: short-term and long-term. The short-term action potential of an information system is focussed upon individual BAT transactions, whereas the long-term action potential of an information system is concerned with the flow of successive BAT transactions. The short-term action potential of an information system is concerned with the rule-governed behavior of that information system within the scope of an individual BAT transaction, and draws heavily upon the information system's short-term action memory. Although primarily rule-governed, each individual BAT transaction will necessarily be influenced by and will strengthen or weaken the evolving business relationships captured in long-term action memory. The long-term action potential of an information system is not essentially rule-governed but is primarily influenced by its long-term action memory and hence the continual evolution of on-going business relationships spanning across multiple BAT transactions. The long-term action potential of an information system forms the behavioral context of successive individual BAT transactions.

2.5 A Language Action Perspective on Information Systems Development

Information systems reflect and support both individual BAT transactions and the flow across successive BAT transactions, as described in section 2.4. It is this combination of individual BAT transactions and their coordinated successive flow that constitutes the reflection of and support for an organization's business processes. The development of information systems, then, must be focussed upon the determination of both the short-term and long-term action potential as well as the short-term and long-term action memory necessary for that information system to successfully reflect (support) those business processes.

The requirements for an information system capture those aspects of an organization's business processes that the information system must support. The determination of those requirements, and their reflection in an information system, are themselves business processes, the participants in which are the information systems developers (ISD) and the information systems users. Those business processes, like any business process, will consist both of individual (short-term) BAT transactions and the long-term flow across successive BAT transactions resulting from an evolving business relationship between ISD and users. Adopting this Language Action Perspective on information systems development implies that the determination of the requirements for an information system is itself a collaborative act following the BAT transaction pattern, so that information systems requirements are not dictated, rather they are negotiated between ISD and information system users. That is, the Language Action Perspective leads to an inherently participatory rather than exclusionary and dictatorial model of information systems development, see for example [Ehn, 1988].

There are, then, two simultaneous levels of long-term business relationship each reflected in short-term BAT transactions. The first and principle level for the organization consists of those business relationships and corresponding BAT transactions which exist between the organization and its stakeholders in fulfillment of the organization's long-term business goals. These stakeholders are the actors who have an interest in and are affected by the organization's business processes. It is these stakeholders with whom the organization collaborates in fulfillment of its own business goals. Such *Business Process Stakeholders* are typically external (customers, suppliers, regulatory bodies, etc) rather than internal (employees) to the organization. This first level, then, is concerned with the primary business processes of the organization – the processes by which it does business on a day to day basis. The second level of application of the BAT transaction pattern, on the other hand, consists of business processes concerned with the development of information systems that support those primary business processes. Again, each business process at this secondary level encompasses a succession of individual short-term BAT transactions with a long-term coordinated flow across them. The aim of this second level is to track, and thus support, on-going change in the organization's primary-level business processes. The *Information System (IS) Stakeholders* participating in this second level tend to be more internal to the organization (ISD and information system users) than external. There will, of course, be some overlap between the two groups of stakeholders, but we still find it useful to distinguish between the two groups.

Since information system requirements are determined by a shared understanding of the primary-level business processes they support, an essential first step in the (secondary-level) development of information systems is to uncover the nature of those (primary-level) business processes. Goldkuhl and Lyytinen observe that uncovering an organization's business processes is essentially a re-constructive process [Goldkuhl and Lyytinen, 1984] wherein intuitive (tacit) knowledge is made explicit in terms of the business processes to which that knowledge contributes. Once agreement has been reached with respect to the nature of business processes, it becomes possible to identify requirements for an information system supporting those business processes. An information system requirements document, we believe, is not a definitive mandate of expected system behavior, rather it simply records the current position of negotiations between IS stakeholders.

One commonly proposed approach to identifying information system requirements is for a systems analyst to interview a selection of IS Stakeholders, and attempt to accommodate all of their divergent perspectives, usually by making certain compromises to resolve any conflicts that arise. The analyst's proposal document is then taken back for review by the IS stakeholders who either agree with the document or propose changes that trigger another iteration of this process. This approach is consistent with [Goldkuhl and Lyytinen, 1984], since there is a participatory process between the analyst and the user. Note however, that this is essentially a one-on-one process of participation, where the analyst is performing conflict resolution across interviews to accommodate divergent perspectives and expectations. Lind and Seigerroth [Lind and Seigerroth, 1999] highlight reservations about interview-based elicitation, since it has been found that conflicts are best resolved by negotiation directly between the stakeholders themselves. That is, the participatory requirements negotiation process is a process whereby the information system users reach mutual commitment to a set of information system requirements with which they are all satisfied. Conflict resolution, then, becomes the domain of the users themselves, rather than the dictate of the analyst. The role of the analyst (or rather, the role of a facilitator), then, becomes primarily one of catalyst for such negotiation. This role calls for a senior and, where possible, impartial facilitator with strong inter-personal and conflict-resolution skills, as well as an understanding of both the business and the technical issues likely to emerge during negotiations.

Stakeholders often recognize seemingly diverse models of their business as expressing essentially the same concept. This openness to the perspectives of other stakeholders results in surprisingly rapid agreement among the stakeholders during the participatory process of negotiation. In other words, by bringing multiple stakeholder positions to the negotiating table, we are able to reach relatively rapid commitment to a mutually acceptable and shared understanding of the business. It is imperative that these negotiations result in a shared commitment to a common and mutually satisfactory perception of the primary-level business processes so that the information system developed in support of those business processes will be mutually satisfactory to all IS stakeholders.

2.6 Information System Development Workshops

The purpose of a participatory requirements determination process is to first negotiate and commit to a mutual business *ontology* shared by all stakeholders, and then to negotiate from this ontology a shared commitment to requirements for an information system which supports this ontology. An ontology [Uschold and Gruninger, 1996] is a shared understanding committed to by a group of actors as a common perception of some domain of interest. The analyst, then, acts as a facilitator in participatory *ontology workshops* wherein IS stakeholders (and possibly Business Process Stakeholders) follow the BAT transaction pattern to negotiate, commit to, and produce a shared understanding of the business processes upon which the information system requirements will be based. This shared ontology is then taken to participatory *requirements workshops* - wherein information systems requirements in support of these business processes are negotiated and committed to. The resulting information system requirements are then taken to a third wave of workshops – *development workshops* – wherein the requirements are realized in collaboratively designing, prototyping, and implementing information systems which support the organization’s business processes. In general, these three types of participatory workshop tend to be followed in the order described, but there is an inherent need for overlap and iteration between them, hence their performance is unlikely to be purely sequential. Consequently, each workshop will frequently result in realizations fed back to earlier workshop phases, thus triggering further iterations (waves) of the development process.

Underpinning our strategy of LAP-oriented information systems development, then, are three sets of participatory workshops (ontology, requirements, and development). Each workshop is a relatively short-term application of the BAT transaction pattern. Each workshop begins with a search for appropriate stakeholders (affected actors) to bring to the table. The stakeholders in a workshop then undertake a process of negotiation to reach a shared commitment to some common output. The final part of each workshop is where the participants actually produce that output in fulfillment of the negotiated commitment. The type of output produced is dependent upon the type of workshop.

In addition to each individual workshop following the BAT transaction pattern, there is also an overall application of the BAT transaction pattern spanning across all of these workshops. This is the BAT transaction that captures and co-ordinates the complete information system development process, as depicted in Figure 2 below. Here, the first two types of workshop (ontology and requirements workshops) cover the process of reaching mutual commitment to a shared understanding of the business processes that will be supported by the information system. This shared understanding is utilized to realize an information system in the final fulfillment phase (the development workshops). Note that the IS development BAT transaction is itself part of a longer-term BAT transaction (not shown in Figure 2) wherein there is a continual evolution of an organization’s information systems (through an on-going succession of IS development transactions). This longer-term BAT transaction aligns the behavior of the information system with the continually changing support needs of the organization’s continually changing business processes.

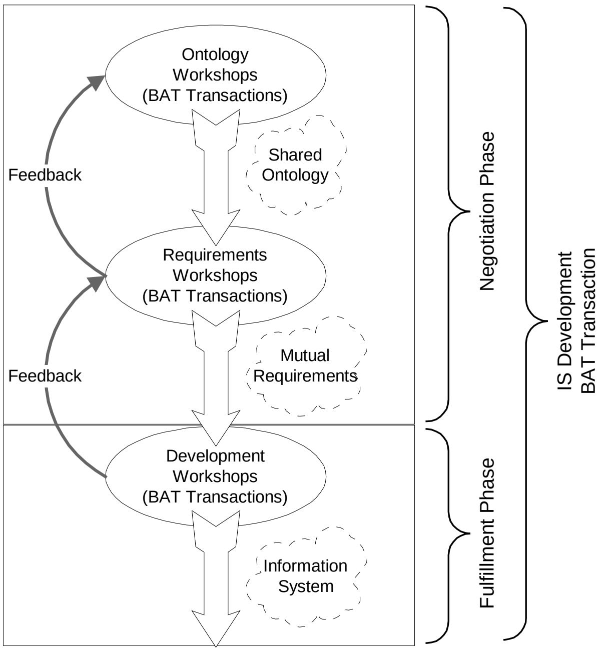

Figure 2: LAP-Oriented IS Development

Participants in all workshops construct models of their negotiated commitments and their workshop output. This both aids the negotiation process and provides a progressive and permanent record of the mutual commitments emerging from those negotiations. Since workshop participants have different backgrounds, agendas, and biases, there is a tendency for them to view the same information through diverse conceptual lenses. In particular, any ambiguities in models are likely to lead to different interpretations and hence different understandings about what is being negotiated and committed to. It is imperative that, for workshops to be effective, misunderstandings about what is being discussed, and progressively committed to, are eliminated. Consequently, the evolving models that record progressively negotiated commitments must be presented without ambiguity throughout the workshops. Natural language has long been recognized as both verbose and potentially ambiguous and hence is probably not the best choice for expressing commitments. In response, a variety of graphical notations have been proposed which are both compact and (to varying degrees) less ambiguous than natural language. The graphical notations traditionally associated with BAT are Action Diagrams and Process Diagrams [Lind and Goldkuhl, 1997]. However, we are not in the habit of dictating specific modeling notations, and hence workshops participants are free to adopt any notations they feel are appropriate to them as a group given their shared understanding of the needs of their own particular problem situation.

3. A Language Action Perspective on Legacy Information Systems

3.1 Legacy Systems Resist Reflection of Business Change

In the introduction to this paper, it was noted that legacy systems are simultaneously business assets and business liabilities. They are assets in that, through years of debugging effort they have grown to reflect essential tacit knowledge - underpinning many of the organization's business processes - the origins and conscious awareness of which are lost in the history of the organization. They are liabilities, however, in that it has become increasingly difficult to adapt them to reflect ongoing business process change [Brodie and Stonebraker, 1995].

3.1.1 Legacy Systems as Business Assets

An organization is rarely in the fortunate position of being completely reflective about its collaborations and contractual negotiations and commitments with stakeholders. Hence, the results of such collaborations (and the commitments emerging from them) are frequently not recorded explicitly, rather they are often implicit in the business processes that have evolved to fulfill them. That is, the contracts negotiated with stakeholders are often absorbed into an organization's business processes without the generation of an explicit audit trail to explain why (and even how) that absorption occurred. In other words, knowledge of those contracts is frequently not explicit, rather it is implicit in the business processes themselves. The existence of these contracts is part of what is termed the *tacit knowledge* of an organization.

The tacit knowledge of an organization includes both knowledge of the individual types of BAT transaction in which the organization participates, and also knowledge of the history of the motivation for and evolution of the business relationships which gave rise to and influence those BAT transactions. An information system, if it is to cover this tacit knowledge, then, must reflect not just the individual BAT transactions but also the historically determined business relationships that control the flow of, and influence and are influenced by, those individual BAT transactions. Individual BAT transactions are captured in terms of the information system's short-term action potential and short-term action memory, whereas the influence of business relationships is captured in the information system's long-term action potential and long-term action memory.

The initial requirements for a new information system will typically cover only the business process transactions of which an organization is explicitly aware due to a deliberate recording or a period of reflection. A major hurdle for an organization, then, is that any information system derived from such requirements will only partially support an organization's contractual negotiations and obligations. Contract negotiations and commitments which have not been deliberately recorded and which are not obvious after a period of reflection are unfortunately omitted from the requirements of a new information system, and hence will not be supported by that information system. Consequently, new information systems must undergo a long and painful debugging process as deficiencies are uncovered and then accounted for in terms of failure to support this essential tacit knowledge.

A legacy system is described as an asset, then, since it has undergone a long-term and often painful debugging process resulting from a continual uncovering of tacit knowledge underlying the organization's essential business processes. It is this hard-won reflection of tacit knowledge – in terms of both action potential and action memory - that constitutes the business asset. The real, and justified, fear of organizations is that a new information system replacing a legacy system may well accurately support the knowledge of which a business is explicitly aware, but will miss – and hence be unable to support - the tacit knowledge implicit in the organization's business processes. The fear, then, is that tacit knowledge will not be reflected in a replacement system until that system has undergone a prolonged and painful debugging process after it has already been deployed and its weaknesses are highlighted via its failures to support the user. The organization has already undergone this pain in the long-term debugging of the legacy system, they do not want to go through it again.

3.1.2 Legacy Systems as Business Liabilities

From a Language Action Perspective, business process change is seen as a response to the results of on-going contract negotiation between an organization and its stakeholders in light of a changing business environment. That is, in order to fulfill the obligations of negotiated contracts, a business must evolve its business processes (and the information systems that support them) accordingly. The initial requirements for an information system supporting a business tend to reflect only a *static snapshot* of the organization's contractual negotiations and commitments. Since new contracts are always being negotiated, information systems requirements are by necessity subject to continual change in order to reflect and support those new contracts accurately. Not surprisingly, the current contracts that an organization is obliged to fulfill are typically far removed from those negotiated years or even decades earlier. Consequently, the current requirements of a mature information system are often far removed from the static snapshot taken when that information system was first developed.

Since an information system is, for the most part, designed with its initial requirements in mind, it is no surprise that the designs - the internal structure or architecture - of mature computer systems are typically inappropriate for the substantially changed requirements they face in later life. As contract negotiations and hence requirements evolve, mature information systems tend to be 'hacked' to force their aging designs to reflect these changes. Thus, over time, they grow old dis-gracefully. This results in what we term *accidental architecture* - the architecture of the information system grows, for all intents and purposes, accidentally as a consequence of repeated waves of hacking - emerging more from the ingenuity of the systems developers' ability to 'warp' the code than from any purposeful design. As successive waves of hacking warp the information system further and further, its accidental architecture becomes more and more convoluted, and the intellectual capability of the developers to identify and apply clever hacks diminishes to a point at which the system is declared un-maintainable. That is, the legacy system grows *dis*gracefully, until it reaches a point where it resists reflection of further business process change. Consequently, a legacy system fails to support the evolving processes of a business – either forcing the business to work around rather than with the information system, or even to abandon essential process changes altogether. At this point the system has become a business liability.

3.2 Replacement Systems: Eliminating the Liability, Retaining the Asset

To support a living, learning, organization successfully, we must maintain the reflection of tacit knowledge which makes legacy systems into assets, whilst simultaneously striving to eliminate the resistance to reflecting business process change which makes those same systems into liabilities. A naïve solution is to simply throw the legacy system away - eliminate the business liability - and develop a new replacement system from scratch. Although this approach has obvious appeal it is an extremely risky undertaking. Our industrial collaborator is extremely wary of simply discarding and rewriting their legacy systems due to the understandable fear that many of the requirements implicit in their legacy systems will be missed in the development of a replacement system. As we have seen, it is the embedded support for tacit knowledge that is being referred to when legacy systems are referred to as business assets. When a legacy system is described as an asset, then, it is recognition of the hard-won accuracy of the system's reflection of the historical negotiation of contracts. This hard-won reflection is manifest in the legacy system's current functionality supporting the fulfillment of those contracts well. In our experience, the legacy information systems supporting an organization's tacit knowledge are often the only point of capture of this knowledge. If a replacement system to a legacy system is to support the organization as effectively as the legacy system then it must also reflect all of the tacit knowledge implicit in that legacy system.

3.2.1 Business Process Re-engineering: Throwing out the Assets with the Liabilities

We have seen that simply throwing away legacy systems and rewriting them without appeal to their hard-gained reflection of tacit knowledge is a dangerous game. There are occasions when this is a useful strategy, such as when the organization is redefining itself through, for example, radical business process re-engineering [Hammer, 1990], but in most cases this is not the wisest path to take. Even when the business does wish to completely redefine itself, we believe that business process re-engineering and green-field development without appeal to the existing business and its legacy systems should be approached with great caution. Radical activities like business process re-engineering - which starts with a blank piece of paper and re-thinks the organization without reference to its current behavior – hold a particularly danger. The inherent and deliberate ignorance of current business processes results in an ignorance of currently negotiated contracts and their process of negotiation established with stakeholders. Such contracts and their negotiations are the basis of relationships (goodwill) built up as trust is formed between an organization and its stakeholders, sometimes over partnerships lasting for decades. Ignorance and potentially the implicit rejection of these transactions then implies ignorance and potentially the implicit rejection of these important business relationships. Major business process re-engineering, we feel, must be sensitive to this inherent rejection of established business relationships and participants in business process reengineering must deliberately pursue the re-negotiation of lost contracts underlying key business relationships and explain to stakeholders how the reengineering effort will result in better value for them.

3.2.2 Business Process Evolution: Respecting and Reflecting the Business Assets

Radical business process reengineering certainly has its place, and should be supported wholeheartedly when a business is in dire straits. However, the majority of business process change is, and will probably continue to be, a result of continual and deliberate business process evolution (rather than revolution) to provide best value in on-going contract negotiations. That is, organizational change usually results not from major overhauls as a consequence of large scale process re-engineering efforts, but from the perpetual improvement of existing processes, which respect and take as their starting point currently negotiated contractual obligations and hence embrace established business relationships. Consequently, any information system that replaces a legacy system must respect the business contracts which are already in place (both explicit and implicit) as well as being open to accommodating future contract evolution. The requirements for such replacement systems, then, must be substantially based upon the requirements implicit in the functionality of the legacy systems that they replace. In other words, the BAT perspective teaches us that business relationships are established, maintained, and strengthened through the transactional negotiation of, commitment to, and mutually satisfactory fulfillment of exchange contracts. Unless those transactions are respected in a replacement system, there exists a risk of diminishing the trust inherent in established business relationships and hence weakening the organization’s position with respect to its partners.

3.2.3 Requirements Mining: Capturing the Business Assets

To respect, in a replacement to a legacy system, the established business process transactions underlying ongoing business relationships we first have to understand exactly what those transactions are. Those transactions – or more accurately, support for them – constitute the requirements (both explicit and tacit) for the organization’s current information systems. Since the requirements supported by a legacy system are often implicit in its functionality, rather than explicitly recorded, the determination of those implicit requirements must begin with a process of requirements reconstruction from the functionality of that legacy system. Before a replacement system can be developed, then, the existing legacy system must be *mined* and the requirements –(in terms of the support of primary-level BAT transaction) implicit in that system must be extracted and recorded explicitly. These minded requirements and the BAT transactions they support, then, form the starting point for a process of negotiation between the stakeholders to reach mutual commitment to requirements for the replacement system.

3.2.4 Code Reuse

Another option for legacy system migration is componentization and reuse of actual code from the legacy system directly in the replacement system. It transpires, however, that the appeal of this approach does not lie in any major saving in terms of programming effort. Our industrial collaborator assures us that the actual programming effort involved in developing replacement systems is almost trivial - they claim rates of approximately two thousand lines of high quality code per developer per week - once the requirements are crystal clear. The only advantage of a code reuse strategy, then, is that the requirements captured within that code are themselves then effectively being reused. The real ambition, then, is not code reuse for its own sake but rather the guarantee that requirements embedded in the legacy code are accommodated in the replacement system. Working with our industrial collaborator, we have found code reuse to be a poor way of achieving this.

Code reuse - or rather code scavenging in the case of legacy systems - is fraught with risks. One obvious risk is that the reclaimed code cannot be made to collaborate with new code. This is certainly the case for our collaborator - much of the legacy code is written in incompatible programming languages that will not execute on the desired hardware platforms and operating systems. Perhaps a greater risk, though, is that even if the code can be reused, internally it will still suffer from the accidental architecture we are trying to escape from. That is, the architecture embedded in the componentized legacy system will be propagated to the replacement system. We have seen this result in replacements to legacy systems which are themselves legacy systems - albeit ones running on the latest buzzword compliant technology. In such cases, the term legacy system migration has been taken almost too literally - the legacy system has been migrated to a new legacy system on a new platform.

3.2.5 Purposeful Architecture

A legacy system is a system that is resistant to reflecting business process change [Brodie and Stonebraker, 1995]. Consequently, a desirable replacement for a legacy system - a non-legacy system - is not simply a system using the latest technology but, rather, a system that is not resistant to reflecting business process change. This implies that code reuse for the purposes of stealthy tacit requirements reflection is of dubious merit if it brings with it a reflection of the legacy system's accidental architecture. To accommodate business process change, the replacement system must exhibit a *purposeful architecture* - an architecture constructed with the reflection of business process change in mind - rather than inheriting the accidental architecture of its predecessor. The major merit of a purposeful architecture is that it readies the information system for the controlled absorption of future requirements emerging from ongoing business process change.

4. Participatory Workshops: Capturing, Reflecting, and Nurturing Business Assets

To summarize our position thus far: we have found that the code and the design (architecture) underlying legacy systems are liabilities to the business. The requirements implicit in legacy systems, however, are fundamental business assets – they capture the transactions (as in the BAT transaction framework) underpinning the organization's business processes. It is possible to build upon these assets, and thus reshape business processes to reflect current business challenges and develop the replacement information systems that will support them, by following a series of participatory workshops as described below.

Section 2.6 covered the three types of participatory information systems development workshops. The workshops described in section 2.6 do not apply only to green-field information systems development efforts. They are equally applicable to the development of replacement information systems for legacy systems. The adoption of this workshop-oriented development strategy facilitates within the organization a continual and deliberate process of purposeful reflection upon and controlled response to on-going business process change. To recap, but with an emphasis on legacy systems, the three types of workshop are:

1. *Ontology Workshops* - which focus upon reaching a shared understanding of the domain supported by the legacy system and, starting with that shared understanding, negotiating and commiting to a refactored understanding of the domain to be supported by the replacement system.
2. *Requirements Workshops* - which focus upon a mutual determination of requirements for the replacement system starting with the shared domain understanding resulting from the ontology workshops.
3. *Development Workshops* - which focus upon collaboratively designing, prototyping, and implementing replacement systems according to the requirements that emerge from requirements workshops.

All of the workshops described here follow the BAT transaction pattern – in that stakeholders negotiate, commit to, and produce output based upon their shared understanding of some reality (note that the commitment is to some shared understanding of reality rather than of some absolute reality itself).

4.1 Ontology Workshops

As with green-field development, the first step towards the development of replacements to legacy systems is the negotiation of a shared business ontology. Once ontological commitment has been reached, an organization is then in a position to utilize that business ontology - the shared understanding - as a basis from which to negotiate the requirements of replacement information system supporting that business.

For the development of replacement systems, there are three types of ontology workshop:

- *Gold Mining* – where the legacy systems themselves are mined to uncover and reconstruct the requirements they implement.

- *Business Process Reconstruction* – where the reconstructed legacy system requirements are taken as the starting point in the negotiated reconstruction of the business processes they support and the business processes to which an organization is currently committed. Business process reconstruction is a process of capturing a snapshot of the organization's business processes.
- *Contract Refactoring* – where the reconstructed business processes are refactored (reshaped) to accommodate resolutions to challenges currently facing, and expected to face, the business.

4.1.1 Gold Mining Workshops

We have seen that to eliminate the liability of legacy systems, ambitions for wholesale code reuse and design reuse must be - for the most part - abandoned, since they simply propagate the liability inherent in the legacy system to the replacement system. The business assets inherent in the requirements underlying the legacy systems, however, need to be maintained and reflected in replacement systems.

When developing information systems to replace legacy systems, then, the legacy systems themselves are mined for the capture and explicit articulation of the requirements implicit in their current functionality. These (often-tacit) requirements form the hidden gold that makes legacy systems into business assets. Consequently, we term the process that leads to their discovery *gold mining*. Our legacy system migration strategy, then, involves intensive code-inspection workshops alongside the maintainers of the legacy system.

The purpose of gold mining workshops is to reconstruct and record the requirements implicit in and satisfied by the functionality of the legacy system. This reconstructive process involves a detailed examination of the action potential and action memory (and thus the total action repertoire) of the legacy system in order to uncover exactly what the legacy system is doing. In essence this process is focused upon identifying the collaborative acts that the information system itself performs either as an agent working in isolation or as a tool in support of human collaborative acts [Goldkuhl and Ågerfalk, 1998b]. This is a time consuming, and highly iterative task, but essential to obtaining an explicit record of the painfully negotiated requirements (in support of business knowledge) which make the legacy system a business asset. Gold mining workshops will primarily be the domain of IS Stakeholders, and in particular systems development and maintenance staff. Business Process Stakeholders will, of course, need to be consulted where the legacy system itself can only be understood with the support of business expertise. The result of gold mining is a deep understanding of the individual collaborations in which the legacy system participates. There is, at this stage, little understanding of the relationship that exists between these individual acts and, more particularly, the short term and long term BAT transactions they support. This relationship to business process support is the focus of the next wave of workshops (business process reconstruction).

4.1.2 Business Process Reconstruction Workshops

Once the gold mining workshop participants have reached agreement that the captured requirements constitute an acceptable reflection of the legacy system's functionality, a second round of ontology workshops is held with both IS and Business Process Stakeholders. During this second round – the business process reconstruction workshops - the collaborative acts expressed within the reconstructed legacy system requirements are taken as the basis for a negotiated reconstruction of the underlying business processes (BAT transactions) that are supported by the legacy system. This then acts as a catalyst in further negotiation of a shared commitment to a mutual understanding of the organization's current business process commitments (including those not adequately supported by the legacy system). This type of workshop closely mirrors the reconstruction seminars described in [Lind and Seigerroth, 1999], wherein commitments are reached to a negotiated mutual understanding of a snapshot of an organization's current business processes.

Process reconstruction workshops often, but not always, essentially follow a process of refining the legacy system requirements, reorganizing them to reflect abstractions common to the stakeholders, accommodating current business processes not adequately supported by the legacy system, and renaming areas where more appropriate business terms exist. In effect, this is a refiltering of the requirements and accommodation of current business processes from the stakeholders' perspectives – expressed in stakeholder terms and using their abstractions rather than the terms and abstractions of the legacy systems implementers.

The aim of business process reconstruction workshops is to capture both the short-term individual BAT transactions, and also the long-term BAT transactions (reflecting and respecting business relationships) which influence and are influenced by them. This, then, results in a more holistic view of the organization and its business processes – capturing the role of the information system and also human actors in both the individual BAT transactions and the wider context within which they participate. The collaborative acts captured in the gold mining workshops are here related to each other as contributing to the support of current business processes (BAT transactions).

4.1.3 Ontology Refactoring Workshops

The reconstructed business processes, resulting from the previous two ontology workshops, reflect the historical negotiation of contracts to which the business is currently committed. It is these contracts which underpin the relationships the organization has formed with its stakeholders and thus constitute assets which bring real value to the business. The third and final round of ontology workshops, ontology refactoring workshops, take these reconstructed business processes as a statement of where the business currently is.

A prerequisite to undertaking ontology refactoring workshops is that the organization has reflected upon and reached a shared understanding of its goals and the current problems facing the business which prevent those goals from being realized [Goldkuhl and Röstlinger, 1993]. Possible solutions to the problems that the business is currently facing are modeled and discussed in the ontology refactoring workshops. In light of these discussions, a newly negotiated business ontology is constructed which is a transformation (a reshaping, or refactoring) of the current business processes - respecting the transactional contract negotiations, commitments, and fulfillments embedded within them - with an eye on openness to change to accommodate the negotiated problem resolutions. These ontology refactoring workshops are necessarily a cooperative act between Business Process and IS Stakeholders, since it is important to ensure that the result of these workshops is both relevant to the current (and anticipated future) challenges of the business and also realistic in terms of support by information systems.

4.2 Requirements Workshops

Requirement workshops form a second wave of workshops following earlier ontology workshops. To recap, participants in the ontology workshops negotiate a shared understanding of where the business currently is, reflect upon solutions to problems currently faced by the business, and negotiate a shared commitment to a transformed business ontology. This new business ontology is the basis of a managed change plan for the business. A replacement information system, then, should not mirror the legacy system that precedes it, but rather should support the new ontology and the planned business change.

Requirements workshops take the new business ontology and the corresponding change plan as the starting point for the negotiation of requirements for the supporting information systems. A particular emphasis of requirements workshops is the determination of the boundaries of the information system within the business ontology; asking which areas it will support and which roles it will fill in the support of those business areas. This, then, is a process of determining the collaborative acts in which the information system will participate and the action repertoire that realizes that participation. Each of those collaborative acts will be either undertaken in support of human interaction or performed independently by the information system acting as a rule-governed autonomous agent.

It is important to include the developers of the information system in these workshops to get them talking to other IS and Business Process Stakeholders - to understand the system from the position of the stakeholders and the support roles of the information system to the business. This both teases the developers away from a technology-centered solution - emphasizing requirements-first - and encourages an ongoing dialog between the maintainers of the system and those affected by it. Furthermore, since the developers will have a very real understanding of just what is feasible in the realization of an information system, their participation can help to keep any unrealistic information system ambitions of over-enthusiastic stakeholders grounded in what is practical and achievable.

4.3 Development Workshops

The result of requirements workshops, then, is a negotiated commitment to a set of information systems requirements and an on-going dialog between developers and other stakeholders. Developers will then withdraw to their own workshops – a third wave of workshops termed development workshops - where system designs and prototypes are negotiated, committed to, and produced (following the BAT transaction pattern) and ultimately a satisfactory implementation is delivered, using these requirements as a starting point. Naturally, the developers will probably hold further requirements workshops with the other stakeholders where ambiguities are clarified, prototypes are evaluated, and plans altered as necessary [Ehn, 1988].

5. Conclusions

In this paper we have argued that legacy information systems are simultaneously business assets and business liabilities. A legacy system is a business asset in that it captures the tacit knowledge underlying an organization’s business processes. A legacy system is a business liability in that its accidental architecture resists the reflection of on-going business process change. To adequately support a developing business, a replacement information system must maintain and enhance this business asset aspect of a legacy system, whilst also eliminating the liability inherent in the legacy system’s accidental architecture.

We have applied a Language Action Perspective to both the collaborative behavior of an organization, in terms of understanding its business processes, and also to the nature and development of information systems which support those business processes. This Language Action Perspective leads us to a transactional view of business processes following the BAT transaction pattern. The important realization is that business processes fulfill contracts negotiated between and committed to with stakeholders. Since negotiation is perpetual, new contracts emerge continually and, as a consequence, business processes must change to reflect those new contracts. Information systems support business processes, and hence support negotiated contracts. Therefore, we see two levels (primary and secondary) of application of the BAT transaction pattern. The primary level addresses the actual business processes that satisfy the needs of the on-going business relationships between an organization and its stakeholders. This is concerned, then, with the normal day-to-day business activities of the organization. The secondary level is concerned with the negotiation of a shared business ontology in light of an examination of an organization's business processes and the challenges it currently faces, and the development of information systems to support those business processes and accommodate solutions to those challenges.

In the paper we have distinguished between short-term and long-term BAT transactions. Short-term BAT transactions are focused primarily upon individual contract negotiation and fulfillment in collaboration with stakeholders. Long-term BAT transactions, on the other hand are concerned with reflection of and support for developing business relationships, and thus both influence and are strengthened or weakened by the consequences of the short-term BAT transactions. When the only source of capture of the long-term and short-term primary-level BAT transactions (i.e. the organization's business processes) supported by an information system is within the information system itself, and when that information system exhibits an accidental architecture, the system is termed a legacy system.

To manage the legacy system issue we have recommended several waves of participatory workshops, wherein the primary-level BAT transactions implicit in a legacy system are first reconstructed and refactored, and then taken as the basis for the negotiation of requirements for the development of a replacement information system. Participatory workshops are primarily focussed upon:

- Negotiation of mutual ontological commitment with respect to that aspect of the business which is supported by the legacy information system
- Negotiation of mutual commitment to a change plan to resolve outstanding business problems
- Negotiation of mutual commitment to a new ontology on which to base the requirements for, and hence the development of, replacement information systems.

These commitments are invaluable assets, but just as important is that we have now established a process (following the BAT transaction framework) whereby the Business Process Stakeholders and the IS Stakeholders are talking to one another. This process focuses their attention on the area of the business that affects them all and encourages their joint participation in the determination of where that business currently is and where it could - or should - go next. This process should not be considered a one-off effort, but rather should become an integral part of the business (a new set of high-priority business processes) thus encouraging a self-aware, reflective, learning, and reactive organization. Thus, the participatory workshops themselves constitute relatively short-term secondary-level BAT transactions, coordinated by and influencing a longer-term secondary-level BAT transaction (business process) concerned with the continual development of the organization and the information systems supporting it.

A major benefit of participatory workshops, then, is that they lead an organization to think about change before it creeps-up on the business and before it creeps-up on the maintainers of information systems as late and unexpected requirements for urgent system hacks. When an organization manages it change purposefully it is able to determine and accommodate the impact of change to the captured business ontology. The organization is then in a position to inform system developers and maintainers of the anticipated impact of such change on the information systems that support the business. More particularly, when an information system's architecture is structured to mirror, and track changes to, the explicitly recorded business ontology it supports, that information system achieves a state of *proximity*.

Proximity means that a small change to the business will require a parallel small change to the information system that supports it. A larger change to the business - the result of radical business process re-engineering in the most extreme case - may, of course, require a correspondingly large change to the information system. The advantage of proximity is that it eliminates nasty surprises. There is nothing worse than to hear from development staff that "It might sound like a small change, but the system wasn't designed that way so its going to be months of development work - that's if we can even schedule for it." Proximity is the key to what we term *purposeful architecture*, where the information system is designed to mirror the negotiated business model it reflects. Purposeful, as opposed to accidental, architecture is key to eliminating the liability inherent in legacy systems – it ensures that today's newly developed information systems do not become tomorrow's legacy systems. In terms of implementation technologies, proximity can, if so desired, be supported by object oriented and agent based approaches. The key here, though, is not a technological decision - rather it goes back to an on-going commitment by the business to undertake a perpetual process of reflection and negotiation and the explicit capture of the resultant business ontology. Purposeful architecture, then, results from purposeful reflection and navigation of the business. That is, from the installation of and perpetual adherence to a learning process.

Finally, it is our firm belief that this participatory process of active and joint reflection - armed with an explicitly recorded shared understanding of the business - is of major benefit to an organization, irrespective of whether or not a resultant information system is intended. Future business process change can then be the result of purposeful reflection through successive waves of participatory workshops. That is, once a business has set in motion a change management process of participatory business reflection and navigation, we feel very strongly that the momentum needs to be maintained perpetually. Participatory workshops, then, should be seen as a regular and on-going commitment to a purposeful steering of the business.

To build upon the foundations presented in this paper, we envision and are undertaking further research in the following areas:

- Integration between different types of model expressing BAT transactions, and the flow of those models across successive waves of participatory workshops.
- A comparative evaluation of various modeling notations with which to express the BAT transaction models constructed in the participatory workshops.
- A symmetric four-role model, building upon the BAT approach, wherein each participant in a collaboration is simultaneously a customer (with needs) and a supplier (with capacity). Collaborations between two actors may occur when the capacities of each satisfy the needs of the other.
- A direct application of the BAT approach in the implementation of the replacement information systems resulting from the participatory workshops, thus resulting in direct reflection of the proximity between requirements and their realization.

Reference List

Austin, J.L. (1962) *How To Do Things With Words*, Clarendon Press.

Brodie, M.L. and Stonebraker, M. (1995) *Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach*, Morgan Kaufman.

Dietz, J.L.G. (1994) Modelling Business Processes For the Purpose of Redesign. In: *Proceedings of IFIP TC8 Open Conference on Business Process Redesign*, North Holland.

Ehn, P. (1988) Work-Oriented Design of Computer Artifacts, PhD Thesis. Almquist and Wiksell International, Stockholm.

Flores, F. and Ludlow, J.J. (1981) Doing and Speaking in the Office. In: Fick, G. and Spraque Jr., H., (Eds.) *Decision Support Systems: Issues and Challenges*, Pergamon Press.

Goldkuhl, G. (1996) Generic Business Frameworks and Action Modelling. In: *Proceedings of Language/Action Perspective '96*, Springer Verlag.

Goldkuhl, G. (1998a) The Six Phases of Business Processes - Communication and The Exchange of Value. In: *Proceedings of Twelfth Biennial ITS Conference*, Stockholm:

Goldkuhl, G. and Ågerfalk, P. (1998b) Action Within Information Systems: Outline of a Requirements Engineering Method. In: *Fourth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ '98)*, Pisa, Italy.

Goldkuhl, G., Lind, M. and Seigerroth, U. (1998c) *The Language Action Perspective on Communication Modelling: Proceedings of the Third International Workshop*, Jonkoping International Business School.

Goldkuhl, G. and Lyytinen, K. (1982) A Language Action View of Information Systems. In: Ginzberg, M. and Ross, C., (Eds.) *Proceedings of 3rd International Conference on Information Systems*, Ann Arbor.

Goldkuhl, G. and Lyytinen, K. (1984) Information Systems Specification as rule reconstruction. In: Bemelmans, T., (Ed.) *Beyond Productivity: Information Systems Development for Organizational Effectiveness* , North-Holland.

Goldkuhl, G. and Röstlinger, A. (1993) Joint Elicitation of Problems: An Important Aspect of Change Analysis. In: *Proceedings of IFIP w.g. 8.2 Working Conference on Information Systems Development: Human,* *Social, and Organizational Aspects*, Noordwijkerhout, Netherlands.

Habermas, J. (1984) *The Theory of Communicative Action*, Polity Press.

Hammer, M. (1990) Reengineering work: don't automate, obliterate. *Harvard Business Review, July/August*

Lind, M. and Goldkuhl, G. (1997) Reconstruction of Different Business Processes - A Theory and Method Driven Analysis. In: Dietz, J.L.G., (Ed.) *Proceedings of Language/Action Perspective '97*, Veldhoven, The Netherlands.

Lind, M. and Seigerroth, U. (1999) *The Business Flight - A Time- and Resource-Efficient Work Step for Reconstruction of Business Praxis*, To Be Submitted.

Macaulay, S. (1992) Non-Contractual Relations in Business: A Preliminary Study. In: Granovetter, M. and Swedberg, R., (Eds.) Westview Press.

Medina-Mora, R., Winograd, T., Flores, R. and Flores, F. (1992) The Action-Workflow Approach to Workflow Management Technology. In: *Proceedings of 4th Conference on CSCW*, ACM Press.

Morgan, G. (1996) Organizations as Organisms. In: *Images of Organizations*, Sage.

Searle, J.R. (1969) *Speech Acts. An essay in the philosophy of language*, Cambridge University Press.

Searle, J.R. (1979) *Expression and Meaning - Studies in the Theory of Speech Acts*, Cambridge University Press.

Senge, P.M. (1990) *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday.

Uschold, M. and Gruninger, M. (1996) Ontologies: Principles, Methods, and Applications. *The Knowledge Engineering Review* 11, 2

van Reijswoud, V.E. and van der Rijst, N.B.J. (1999) Modelling Business Communication as a Foundation for Business Process Redesign. In: *Proceeding of the 28th International Conference on Systems Science*, IEEE Computer Society Press.

Verharen, E.M. (1997) A Language-Action Perspective on the Design of Cooperative Information Agents, PhD Thesis. KUB, Tilburg.

Winograd, T. (1988) A Language/Action Perspective on the Design of Cooperative Work. In: Greif, I., (Ed.) *Computer Supported Cooperative Work: A Book of Readings*, Morgan Kaufmann.

Winograd, T. and Flores, F. (1987) *Understanding Computers and Cognition: A new foundation for design*, Addison-Wesley.

Wittgenstein, L. (1958) *Philosophical Investigations*, Basil Blackwell.