AN AGENT FOR EMERGENT PROCESS MANAGEMENT

John Debenham
University of Technology, Sydney
PO Box 123, Broadway, NSW 2007, Australic
Email: debenham@it.uts.edu.au

Keywords: Intelligent Agents, Business Process, Agents for Internet Computing.

Abstract: Emergent processes are business processes whose execution is determined by the prior knowledge of the agents involved and by the knowledge that emerges during a process instance. The amount of process knowledge that is relevant to a knowledge-driven process can be enormous and may include common sense knowledge. If a process' knowledge can not be represented feasibly then that process can not be managed; although its execution may be partially supported. In an e-market domain, the majority of transactions, including trading orders, requests for advice and information, are knowledge-driven processes for which the knowledge base is the Internet, and so representing the knowledge is not at issue. Multiagent systems are an established platform for managing complex business processes. What is needed for emergent process management is an intelligent agent that is driven not by a process goal, but by an in-flow of knowledge, where each chunk of knowledge may be uncertain. These agents should assess the extent to which it chooses to believe that the information is correct, and so they require an inference mechanism that can cope with information of differing integrity. An agent is described that achieves this by using ideas from information theory, and by using maximum entropy logic to derive integrity estimates for knowledge about which it is uncertain. Emergent processes are managed by these agents that extract the process knowledge from this knowledge base — the Internet — using a suite of data mining bots. The agents make no assumptions about the internals of the other agents in the system including their motivations, logic, and whether they are conscious of a utility function. These agents focus only on the information in the signals that they receive.

1 INTRODUCTION

Emergent processes are business processes that are not predefined and are ad hoc. These processes typically take place at the higher levels of organisations (Dourish, 1998), and are distinct from production workflows (Fischer, 2003). Emergent processes are opportunistic in nature whereas production workflows are routine. How an emergent process will terminate may not be known until the process is well advanced. The tasks involved in an emergent process are typically not predefined and emerge as the process develops. Those tasks may be carried out by collaborative groups as well as by individuals (Smith and Fingar, 2003) and may involve informal meetings, business lunches and so on. For example, in an e-market context an emergent process could be triggered by "lets try to establish a business presence in Hong Kong". Further, the goal of an emergent process instance may mutate as the instance matures. So unlike "lower-order" processes, the goal of an emergent process instance may not be used as a focus for the management of that instance.

Emergent processes contain "knowledge-driven" sub-processes, but may also contain conventional "goal-driven" sub-processes. A *knowledge-driven process* is guided by its "process knowledge" and "performance knowledge". The *goal* of a knowledge-driven process may not be fixed and may mutate. On the other hand, the management of a *goal-driven process* instance is guided by its goal which is fixed. A multiagent system to manage the "goal-driven" processes is described in (Debenham, 2000). In that system each human user is assisted by an agent which is based on a generic three-layer, BDI hybrid agent architecture. The term individual refers to a user/agent pair. The general business of managing knowledge-driven processes is illustrated in Fig. 1, and will be discussed in Sec. 2.

Process management is an established application area for multi-agent systems (Singh, 2004) although emergent processes are typically handled either manually or by CSCW systems rather than by process management systems. The use of these two technologies is not elegant and presents a barrier to a unified view of emergent process management.

In an experimental e-market, transactions include: trading orders to buy and sell in an e-exchange, single-issue and multi-issue negotiations between two parties, requests for information extracted from market data as well as from news feeds and other Internet data. In this e-market every market transaction is managed as a business process. To achieve this, suitable process management machinery has been developed. To investigate what is "suitable" the essential features of these transactions are related to two classes of process that are at the "high end" of process management feasibility (van der Aalst and van Hee, 2002). The two classes are goal-driven processes and knowledge-driven processes — Sec. 2. The term "business process management" is generally used to refer to the simpler class of workflow processes (Fischer, 2003), although there are notable exceptions using multiagent systems (Singh, 2004).

The agent architecture described extends the simple, offer-exchange, bargaining agent described in (Debenham, 2004). The agent described here is driven by the contents of a knowledge base that represents the agent's world model in probabilistic first-order logic, and manages emergent processes. Each message that the agent receives from another agent reveals valuable information about the sender agent's position. The agent aims to respond with messages that have comparable information revelation. In this way it aims to gain the trust of its opponent. The agent does not necessarily strive to optimize its utility and aims to make informed decisions in an information-rich but uncertain environment.

The emergent process management agent, Π, attempts to fuse the agent interaction with the information that is generated both by and because of it. To achieve this, it draws on ideas from information theory rather than game theory. Π decides what to do — such as what message to send — on the basis of its information that may be qualified by expressions of degrees of belief. Π uses this information to calculate, and continually re-calculate, probability distributions for that which it does not know. One such distribution, over the set of all possible actions, expresses Π's belief in the suitability to herself of the system performing that action. Other distributions attempt to predict the behavior of its opponent, Ω say, — such as what proposals she might accept, and of other unknowns that may effect the process outcome. Π makes no assumptions about the internals of the other agents in the system, including whether they have, or is even aware of the concept of, utility functions. Π is purely concerned with the other agents' behaviors — what they do — and not with assumptions about their motivations. This somewhat detached stance is appropriate for emergent process management in which each agent represents the interests of it owner, whilst at the same time attempting to achieve the social goal of driving the processes towards a satisfactory conclusion.

As with the agent described in (Debenham, 2004), the process management agent described here does not assume that it has a von Neumann-Morgerstern utility function. The agent makes assumptions about: the way in which the integrity of information will decay, and some of the preferences that its opponent may have for some deals over others. It also assumes that unknown probabilities can be inferred using *maximum entropy inference* (MacKay, 2003), *ME*, which is based on random worlds (Halpern, 2003). The maximum entropy probability distribution is "the least biased estimate possible on the given information; i.e. it is maximally noncommittal with regard to missing information" (Jaynes, 1957). In the absence of knowledge about the other agents' decision-making apparatuses the process management agent assumes that the "maximally noncommittal" model is the correct model on which to base its reasoning.

2 PROCESS MANAGEMENT

Following (Fischer, 2003) a *business process* is "a set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships". Implicit in this definition is the idea that a process may be repeatedly decomposed into linked sub-processes until those sub-processes are activities which are atomic pieces of work. [viz (Fischer, 2003) "An activity is a description of a piece of work that forms one logical step within a process."].

A particular process is called a (process) *instance*. An instance may require that certain things should be done; such things are called tasks. A *trigger* is an event that leads to the creation of an instance. The *goal* of an instance is a state that the instance is trying to achieve. The *termination condition* of an instance is a condition which if satisfied during the life of an instance causes that instance to be destroyed whether its goal has been achieved or not. The *patron* of an instance is the individual who is responsible for managing the life of that instance. At any time in a process instance's life, the *history* of that instance is the sequence of prior sub-goals and the prior sequence of knowledge inputs to the instance. The history is "knowledge of all that has happened already".

Three classes of business process are defined in terms of their management properties (ie: in terms of how they may be managed).

- A *task-driven process* has a unique decomposition into a — possibly conditional — sequence of activities. Each of these activities has a goal and is associated with a task that "always" achieves this goal. Production workflows are typically task-driven processes.
- A *goal-driven process* has a process goal, and achievement of that goal is the termination condition for the process. The process goal may have various decompositions into sequences of sub-goals where these sub-goals are associated with (atomic) activities and so with tasks. Some of these sequences of tasks may work better than others, and there may be no way of knowing which is which (Smith and Fingar, 2003). A task for an activity may fail outright, or may be otherwise ineffective at achieving its goal. In other words, failure is a feature of goal-driven processes. If a task fails then another way to achieve the process goal may be sought.
- A *knowledge-driven process* may have a process goal, but the goal may be vague and may mutate (Dourish, 1998). Mutations are determined by the process patron, often in the light of knowledge generated during the process. At each stage in the performance of a knowledge-driven process the "next goal" is chosen by the process patron; this choice is made using general knowledge about the context of the process — called the process knowledge. The process patron also chooses the tasks to achieve that next goal; this choice may be made using general knowledge about the effectiveness of tasks — called the performance knowledge. So in so far as the process goal gives direction to goal-driven — and task-driven — processes, the process knowledge gives direction to knowledge-driven processes. The management of knowledge-driven processes is considerably more complex than the other two classes of process. But, knowledge-driven processes are "not all bad" — they typically have goal-driven sub-processes which may be handled in conventional way. A simplified view of knowledge-driven process management is shown in Fig. 1.

Managing knowledge-driven processes is rather more difficult than goal-driven processes, see Fig. 1. The complete representation, never mind the maintenance, of the process knowledge may be an enormous job. But the capture of at least some of the knowledge generated during a process instance may not be difficult if the tasks chosen used virtual documents such as workspace technology, for example. Some performance knowledge is not difficult to capture, represent and maintain. For example, measurements of how long another agent took to complete a sub-process can be very useful. So in the system described here, the process knowledge is left in the heads of the patron or nominated delegates, and the performance knowledge is captured by the system. The initial selection of the process goal is performed by the patron, and so this action is completely unsupported by the system, see Fig. 1. The possible subsequent mutation of the process goal is performed by the agent using the process knowledge, see Fig. 1. Task selection is supported by the agent for e-market processes which can, for example, be given authority to withdraw a bid from two separate auctions and to negotiate for a package of goods from a single supplier. In this way the system provides considerable assistance in the management of knowledge-driven processes. Further, if a now-goal is associated with a goal-driven, or task-driven, sub-process then the management system is given full responsibility for the management of that sub-process.

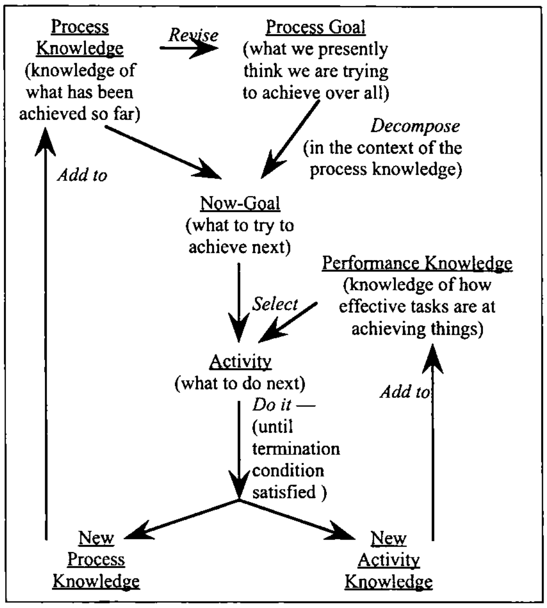

Figure 1: Knowledge-driven process management

3 EMERGENT PROCESS AGENT

Π operates in an information-rich environment that includes the Internet. The integrity of Π's information, including information extracted from the Internet, will decay in time. The way in which this decay occurs will depend on the type of information, and on the source from which it is drawn. Little appears to be known about how the integrity of real information, such as news-feeds, decays, although the effect of declining integrity has been analyzed. For example, (Bernhardt and Miao, 2004) considers how delays in the acquisition of trading data effect trading outcomes.

One source of Π's information is the signals received from Ω. These include offers from Ω to Π, the acceptance or rejection by Ω of Π's offers, and claims that Ω sends to Π. This information is augmented with sentence probabilities that represent the strength of Π's belief in its truth. If Ω rejected Π's offer of $8 two days ago then what is Π's belief now in the proposition that Ω will accept another offer of $8 now? Perhaps it is around 0.1. A linear model is used to model the integrity decay of these beliefs, and when the probability of a decaying belief approaches 0.5[1] the belief is discarded. The model of decay could be exponential, quadratic or what ever.

3.1 Interaction Protocol

A *deal* is a pair of commitments $\delta_{\Pi:\Omega}(\pi, \omega)$ between an agent Π and an opponent agent Ω, where π is Π's commitment and ω is Ω's commitment. $\mathcal{D} = \{\delta_i\}_{i=1}^{D}$ is the deal set — ie: the set of all possible deals. If the discussion is from Π's point of view then the subscript "$\Pi : \Omega$" may be omitted. These commitments may involve multiple issues and not simply a single issue such as trading price. The set of *terms*, $\mathcal{T}$, is the set of all possible commitments that could occur in deals in the deal set. An agent may have a real-valued *utility* function: $\mathbf{U} : \mathcal{T} \rightarrow \Re$, that induces an ordering on $\mathcal{T}$. For such an agent, for any deal $\delta = (\pi, \omega)$ the expression $\mathbf{U}(\omega) - \mathbf{U}(\pi)$ is called the *surplus* of δ, and is denoted by $\mathbf{L}(\delta)$ where $\mathbf{L} : \mathcal{T} \times \mathcal{T} \rightarrow \Re$. For example, the values of the function $\mathbf{U}$ may expressed in units of money. It may not be possible to specify the utility function either precisely or with certainty.[2] This is addressed in Sec. 4 where a predicate $\Omega Acc(.)$ represents the acceptability of a deal to Ω.

The agents communicate using sentences in a first-order language $\mathcal{C}$. This includes the exchange, acceptance and rejection of offers. $\mathcal{C}$ usual trading predicates including the following: *Offer*(δ), *Accept*(δ), *Reject*(δ), *Bid*(δ) and *Quit*(.), where *Offer*(δ) means "the sender is offering you a deal δ", *Accept*(δ) means "the sender accepts your deal δ", *Reject*(δ) means "the sender rejects your deal δ", *Bid*(δ) means "the sender submits the bid δ" and *Quit*(.) means "the sender quits — the negotiation ends".

[1]A sentence probability of 0.5 represents null information, ie: "maybe, maybe not".

[2]The often-quoted oxymoron "I paid too much for it, but its worth it." attributed to Samuel Goldwyn, movie producer, illustrates that intelligent agents may negotiate with uncertain utility.

3.2 Agent Architecture

Π uses the language $\mathcal{C}$ for external communication, and the language $\mathcal{L}$ for internal representation. Two predicates in $\mathcal{L}$ are: $\Pi Acc(.)$ and $\Omega Acc(.)$. The proposition $(\Pi Acc(\delta) \mid \mathcal{I}_t)$ means: "Π will be comfortable accepting the deal δ given that Π knows information $\mathcal{I}_t$ at time t". The idea is that Π will accept deal δ if $\mathbf{P}(\Pi Acc(\delta) \mid \mathcal{I}_t) \geq \alpha$ for some threshold constant α. The precise meaning that Π gives to $\Pi Acc(.)$ is described in Sec. 4. The proposition $\Omega Acc(\delta)$ means "Ω is prepared to accept deal δ". The probability distribution $\mathbf{P}(\Omega Acc(.))$ is estimated in Sec. 5.

Each incoming message M from source S received at time t is time-stamped and source-stamped, $M_{[S,t]}$, and placed in an *in box*, $\mathcal{X}$, as it arrives. Π has an *information repository* $\mathcal{I}$, a *knowledge base* $\mathcal{K}$ and a *belief set* $\mathcal{B}$. Each of these three sets contains statements in a first-order language $\mathcal{L}$. $\mathcal{I}$ contains statements in $\mathcal{L}$ together with sentence probability functions of time. $\mathcal{I}_t$ is the state of $\mathcal{I}$ at time t and may be inconsistent. At some particular time t, $\mathcal{K}_t$ contains statements that Π believes are true at time t, such as $\forall x(Accept(x) \leftrightarrow \neg Reject(x))$. The belief set $\mathcal{B}_t = \{\beta_i\}$ contains statements that are each qualified with a *given sentence probability*, $\mathbf{B}(\beta_i)$, that represents Π's belief in the truth of the statement at time t. The distinction between the knowledge base $\mathcal{K}$ and the belief set $\mathcal{B}$ is simply that $\mathcal{K}$ contains unqualified statements and $\mathcal{B}$ contains statements that are qualified with sentence probabilities. $\mathcal{K}$ and $\mathcal{B}$ play different roles in the method described in Sec. 3.3; $\mathcal{K}_t \cup \mathcal{B}_t$ is required by that method to be consistent.

Π's actions are determined by its "strategy". A *strategy* is a function $\mathbf{S} : \mathcal{K} \times \mathcal{B} \rightarrow \mathcal{A}$ where $\mathcal{A}$ is the set of actions. At certain distinct times the function $\mathbf{S}$ is applied to $\mathcal{K}$ and $\mathcal{B}$ and the agent does something. The set of actions, $\mathcal{A}$, includes sending *Offer*(.), *Accept*(.), *Reject*(.), *Quit*(.) messages and claims to Ω. The way in which $\mathbf{S}$ works is described in Secs. 5. Two "instants of time" before the $\mathbf{S}$ function is activated, an "import function" and a "revision function" are activated. The import function $\mathbf{I} : (\mathcal{X} \times \mathcal{I}_{t^-}) \rightarrow \mathcal{I}_t$ clears the in-box, using its "import rules". An import rule takes a message M, written in language $\mathcal{C}$, and from it derives sentences written in language $\mathcal{L}$ to which it attaches decay functions, and adds these sentences together with their decay functions to $\mathcal{I}_{t^-}$ to form $\mathcal{I}_t$. These decay functions are functions of the message type, the time the message arrived and the source from which it came — an illustration is given below. An *import rule* has the form: $\mathbf{P}(S \mid M_{[\Omega,t]}) = f(M, \Omega, t) \in [0, 1]$, where S is a statement, M is a message and f is the decay function. Then the belief revision function $\mathbf{R} : \mathcal{I}_{t^-} \rightarrow (\mathcal{I}_t \times \mathcal{K}_t \times \mathcal{B}_t)$ deletes any statements in $\mathcal{I}_{t^-}$ whose sentence probability functions have a value that is $\approx$ 0.5 at time t. From the remaining statements $\mathbf{R}$ selects a consistent set of statements and instantiates their sentence probability functions to time t, and places the unqualified statements from that set in $\mathcal{K}_t$ and the qualified statements, together with their sentence probabilities, in $\mathcal{B}_t$.

An example now illustrates the ideas in the previous paragraph. Suppose that the predicate $\Omega Acc(\delta)$ means that "deal δ is acceptable to Ω". Suppose that Π is attempting to trade a good "g" for cash. Then a deal $\delta(\pi, \omega)$ will be $\delta(g, x)$ where x is an amount of money. If Π assumes that Ω would prefer to pay less than more then $\mathcal{I}_t$ will contain: ι_0 : $(\forall gxy)((x \geq y) \rightarrow (\Omega Acc(g, x)) \rightarrow \Omega Acc(g, y))$. Suppose Π uses a simple linear decay for its import rules: $f(M, \Omega, t_i) = \mathit{trust}(\Omega) + (0.5 - \mathit{trust}(\Omega)) \times \frac{t - t_i}{\mathit{decay}(\Omega)}$, where $\mathit{trust}(\Omega)$ is a value in $[0.5, 1]$ and $\mathit{decay}(\Omega) > 0$.[3] $\mathit{trust}(\Omega)$ is the probability attached to S at time $t = t_i$, and $\mathit{decay}(\Omega)$ is the time period taken for $\mathbf{P}(S)$ to reach 0.5 when S is discarded. Suppose at time $t = 7$, Π receives the message: $\mathit{Offer}(g, \$20)_{[\Omega, 7]}$, and has the import rule: $\mathbf{P}(\Omega Acc(g, x) \mid \mathit{Offer}(g, x)_{[\Omega, t_i]}) = 0.8 - 0.025 \times (t - t_i)$, ie: *trust* is 0.8 and *decay* is 12. Then, in the absence of any other information, at time $t = 11$, $\mathcal{K}_{t_{11}}$ contains ι_0 and $\mathcal{B}_{t_{11}}$ contains $\Omega Acc(g, \$20)$ with a sentence probability of 0.7.

Π uses three things to make offers: an estimate of the likelihood that Ω will accept any offer [Sec. 5], an estimate of the likelihood that Π will, in hindsight, feel comfortable accepting any particular offer [Sec. 4], and an estimate of when Ω may quit and leave the negotiation — see (Debenham, 2004). Π supports its negotiation with claims with the aim of either improving the outcome — reaching a more beneficial deal — or improving the process — reaching a deal in a more satisfactory way.

3.3 Random worlds

Let $\mathcal{G}$ be the set of all positive ground literals that can be constructed using the predicate, function and constant symbols in $\mathcal{L}$. A *possible world* is a valuation function $\mathcal{V} : \mathcal{G} \rightarrow \{\top, \bot\}$. $\mathcal{V}$ denotes the set of all possible worlds, and $\mathcal{V}_{\mathcal{K}}$ denotes the set of possible worlds that are consistent with a knowledge base $\mathcal{K}$ (Halpern, 2003).

A *random world* for $\mathcal{K}$ is a probability distribution $\mathcal{W}_{\mathcal{K}} = \{p_i\}$ over $\mathcal{V}_{\mathcal{K}} = \{\mathcal{V}_i\}$, where $\mathcal{W}_{\mathcal{K}}$ expresses an agent's degree of belief that each of the possible worlds is the actual world. The *derived sentence probability* of any $\sigma \in \mathcal{L}$, *with respect to* a random world $\mathcal{W}_{\mathcal{K}}$ is $(\forall \sigma \in \mathcal{L})$:

$$\mathbf{P}_{\mathcal{W}_{\mathcal{K}}}(\sigma) \triangleq \sum_n \{ p_n : \sigma \; is \top in \; \mathcal{V}_n \} \qquad (1)$$

A random world $\mathcal{W}_{\mathcal{K}}$ is *consistent* with the agent's beliefs $\mathcal{B}$ if: $(\forall \beta \in \mathcal{B})(\mathbf{B}(\beta) = \mathbf{P}_{\mathcal{W}_{\mathcal{K}}}(\beta))$. That is, for each belief its derived sentence probability as calculated using Eqn. 1 is equal to its given sentence probability.

The *entropy* of a discrete random variable X with probability mass function $\{p_i\}$ is (MacKay, 2003): $\mathbf{H}(X) = -\sum_n p_n \log p_n$ where: $p_n \geq 0$ and $\sum_n p_n = 1$. Let $\mathcal{W}_{\{\mathcal{K}, \mathcal{B}\}}$ be the "maximum entropy probability distribution over $\mathcal{V}_{\mathcal{K}}$ that is consistent with $\mathcal{B}$". Given an agent with $\mathcal{K}$ and $\mathcal{B}$, its *derived sentence probability* for any sentence, $\sigma \in \mathcal{L}$, is:

$$(\forall \sigma \in \mathcal{L}) \mathbf{P}(\sigma) \triangleq \mathbf{P}_{\mathcal{W}_{\{\mathcal{K}, \mathcal{B}\}}}(\sigma) \qquad (2)$$

Using Eqn. 2, the derived sentence probability for any belief, β_i, is equal to its given sentence probability. So the term *sentence probability* is used without ambiguity.

If X is a discrete random variable taking a finite number of possible values $\{x_i\}$ with probabilities $\{p_i\}$ then the *entropy* is the average uncertainty removed by discovering the true value of X, and is given by $\mathbf{H}(X) = -\sum_n p_n \log p_n$. The direct optimization of $\mathbf{H}(X)$ subject to a number, θ, of linear constraints of the form $\sum_n p_n g_k(x_n) = \overline{g}_k$ for given constants $\overline{g}_k$, where $k = 1, \ldots, \theta$, is a difficult problem. Fortunately this problem has the same unique solution as the *maximum likelihood problem* for the Gibbs distribution (Pietra et al., 1997). The solution to both problems is given by:

$$p_n = \frac{exp\Big(-\sum_{k=1}^{\theta} \lambda_k g_k(x_n)\Big)}{\sum_m exp\Big(-\sum_{k=1}^{\theta} \lambda_k g_k(x_m)\Big)} \qquad (3)$$

$n = 1, 2, \cdots$ where the constants $\{\lambda_i\}$ may be calculated using Eqn. 3 together with the three sets of constraints: $p_n \geq 0$, $\sum_n p_n = 1$ and $\sum_n p_n g_k(x_n) = \overline{g}_k$. The distribution in Eqn. 3 is known as *Gibbs distribution*.

4 SUITABILITY OF AN ACTION

The proposition $(\Pi Acc(\delta) \mid \mathcal{I}_t)$ was introduced in Sec. 3.2. This section describes how the agent estimates its beliefs of whether this proposition is true for various δ.

[3] In this example, the value for the probability is given by a linear decay function that is independent of the message type, and *trust* and *decay* are functions of Ω only. There is scope for using learning techniques to refine the *trust* and *decay* functions in the light of experience.

4.1 An Exemplar Application

An exemplar application follows. Π is placing bids in an e-market attempting to purchase of a particular second-hand motor vehicle, with some period of warranty, for cash. So the two issues in this negotiation are: the period of the warranty, and the cash consideration. A deal δ consists of this pair of issues, and the deal set has no natural ordering. Suppose that Π wishes to apply *ME* to estimate values for: $\mathbf{P}(\Omega Acc(\delta))$ for various δ. Suppose that the warranty period is simply $0, \cdots, 4$ years, and that the cash amount for this car will certainly be at least \$5,000 with no warranty, and is unlikely to be more than \$7,000 with four year's warranty. In what follows all price units are in thousands of dollars. Suppose then that the deal set in this application consists of 55 individual deals in the form of pairs of warranty periods and price intervals: { (w, [5.0, 5.2)), (w, [5.2, 5.4)), (w, [5.4, 5.6)), (w, [5.6, 5.8), (w, [5.8, 6.0)), (w, [6.0, 6.2)), (w, [6.2, 6.4)), (w, [6.4, 6.6)), (w, [6.6, 6.8)), (w, [6.8, 7.0)), (w, [7.0, ∞)) }, where $w = 0, \cdots, 4$. Suppose that Π has previously received two offers from Ω. The first is to offer 6.0 with no warranty, and the second to offer 6.9 with one year's warranty. Suppose Π believes that Ω still stands by these two offers with probability 0.8. Then this leads to two beliefs: $\beta_1 : \Omega Acc(0, [6.0, 6.2))$; $\mathbf{B}(\beta_1) = 0.8$, $\beta_2 : \Omega Acc(1, [6.8, 7.0))$; $\mathbf{B}(\beta_2) = 0.8$. Following the discussion above, before "switching on" *ME*, Π should consider whether it believes that $\mathbf{P}(\Omega Acc(\delta))$ is uniform over δ. If it does then it includes both β_1 and β_2 in $\mathcal{B}$, and calculates $\mathcal{W}_{\{\mathcal{K},\mathcal{B}\}}$ that yields estimates for $\mathbf{P}(\Omega Acc(\delta))$ for all δ. If it does not then it should include further knowledge in $\mathcal{K}$ and $\mathcal{B}$. For example, Π may believe that Ω is more likely to bid for a greater warranty period the higher her bid price. If so, then this is a multi-issue constraint, that is represented in $\mathcal{B}$, and is qualified with a sentence probability.

4.2 Estimation of Beliefs

Here, agent, Π, is attempting to buy a second-hand motor vehicle with a specific period of warranty as described in Sec. 4.1. This section describes how Π estimates: $\mathbf{P}(\Pi Acc(\delta) \mid \mathcal{I}_t)$. This involves the introduction of four predicates into the language $\mathcal{L}$: *Me*(.), *Suited*(.), *Good*(.) and *Fair*(.).

General information is extracted from the World Wide Web using special purpose bots that import and continually confirm information. These bots communicate with Π by delivering messages to Π's in-box $\mathcal{X}$ using predicates in the communication language $\mathcal{C}$ in addition to those described in Sec. 3.1. These predicates include *IsGood*(Γ, Ω, r), and *IsFair*(Γ, δ, s) meaning respectively that "according to agent Γ, agent Ω is a good person to deal with certainty r", and "according to agent Γ, δ is a fair market deal with certainty s". The continual in-flow of information is managed as described in (Debenham, 2003). As described in Sec. 3.2, import functions are applied to convert these messages into beliefs. For example: $\mathbf{P}(Good(\Omega) \mid IsGood(\Gamma, \Omega, r)_{[\ominus, t_i]}) = f(IsGood, r, \Gamma, t)$, where $Good(\Omega)$ is a predicate in the agents internal language $\mathcal{L}$ meaning "Ω will be a good agent to do business with". Likewise, *IsFair*(.) messages in $\mathcal{C}$ are imported to $\mathcal{I}$ as *Fair*(.) statements in $\mathcal{L}$, where $Fair(\delta)$ means "δ is generally considered to be a fair deal at least".

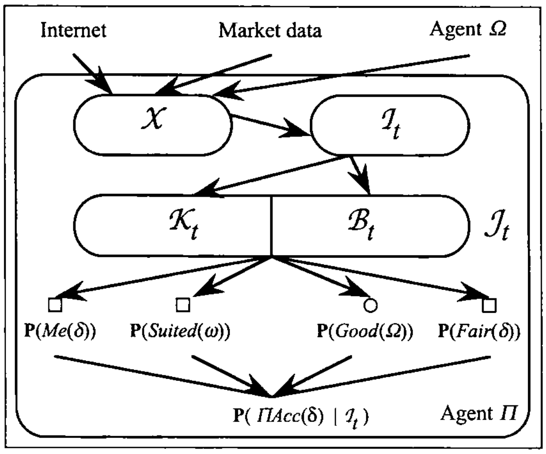

Figure 2: Estimating Π's beliefs

With the motor vehicle application in mind, $\mathbf{P}(\Pi Acc(\delta) \mid \mathcal{I}_t)$ is derived from conditional probabilities attached to four other propositions: $Suited(\omega)$, $Good(\Omega)$, $Fair(\delta)$, and $Me(\delta)$, where $Suited(\omega)$ means "terms ω are perfectly suited to Π's needs", and $Me(\delta)$ means "on strictly subjective grounds, the deal δ is acceptable to Π". These four probabilities are: $\mathbf{P}(Suited(\omega) \mid \mathcal{I}_t)$, $\mathbf{P}(Good(\Omega) \mid \mathcal{I}_t)$, $\mathbf{P}(Fair(\delta) \mid \mathcal{I}_t \cup \{Suited(\omega), Good(\Omega)\})$ and $\mathbf{P}(Me(\delta) \mid \mathcal{I}_t \cup \{Suited(\omega), Good(\Omega)\})$. The last two of these four probabilities factor out both the suitability of ω and the appropriateness of the opponent Ω. The third captures the concept of "a fair market deal" and the fourth a strictly subjective "what ω is worth to Π". The "*Me*(.)" proposition is closely related to the concept of a private valuation in game theory. This derivation of $\mathbf{P}(\Pi Acc(\delta) \mid \mathcal{I}_t)$ from the four other probabilities may not be suitable for assessing other types of deal. For example, in eProcurement some assessment of the value of an on-going relationship with an opponent may be a significant issue. Also, for some low-value trades, the inclusion of *Good*(.) may not be required.

The whole "estimation of beliefs" apparatus is illustrated in Fig. 2.

5 INTERACTION

Π engages in bilateral bargaining with its opponent Ω. Π and Ω each exchange offers alternately at successive discrete times (Kraus, 2001). They enter into a commitment if one of them accepts a standing offer. The protocol has three stages:
1. Simultaneous, initial, binding offers from both agents;
2. A sequence of alternating offers, and
3. An agent quits and walks away from the negotiation.

In the first stage, the agents simultaneously send *Offer*(.) messages to each other that stand for the entire negotiation. These initial offers are taken as limits on the range of values that are considered possible. This is crucial to the method described in Sec. 3.3 where there are domains that would otherwise be unbounded. The exchange of initial offers "stakes out the turf" on which the subsequent negotiation will take place. In the second stage, an *Offer*(.) message is interpreted as an implicit rejection, *Reject*(.), of the opponent's offer on the table. Second stage offers stand only if accepted by return — Π interprets these offers as indications of Ω's willingness to accept — they are represented as beliefs with sentence probabilities that decay in time. The negotiation ceases *either* in the second round if one of the agents accepts a standing offer *or* in the final round if one agent quits and the negotiation breaks down.

To support the offer-exchange process, Π has do two different things. First, it must respond to offers received from Ω — that is described in Sec. 4. Second, it must send offers, and possibly information, to Ω. This section describes machinery for estimating the probabilities $\mathbf{P}(\Omega Acc(\delta))$ where the predicate $\Omega Acc(\delta)$ means "Ω will accept Π's offer δ". In the following, Π is attempting to purchase of a particular second-hand motor vehicle, with some period of warranty, for cash from Ω as described in Sec. 4.1. So a deal δ will be represented by the pair (w, p) where w is the period of warranty in years and $\$p$ is the price.

Π assumes the following two preference relations for Ω, and $\mathcal{K}$ contains:

κ_{11} :
$\forall x, y, z((x < y) \rightarrow (\Omega Acc(y, z) \rightarrow \Omega Acc(x, z)))$

κ_{12} :
$\forall x, y, z((x < y) \rightarrow (\Omega Acc(z, x) \rightarrow \Omega Acc(z, y)))$

As in Sec. 4, these sentences conveniently reduce the number of possible worlds. The two preference relations κ_{11} and κ_{12} induce a partial ordering on the sentence probabilities in the $\mathbf{P}(\Omega Acc(w, p))$ array from the top-left where the probabilities are ≈ 1, to the bottom-right where the probabilities are ≈ 0. There are fifty-one possible worlds that are consistent with $\mathcal{K}$.

Suppose that the offer exchange has proceeded as follows: Ω asked for \$6,900 with one year warranty and Π refused, then Π offered \$5,000 with two years warranty and Ω refused, and then Ω asked for \$6,500 with three years warranty and Π refused. Then at the next time step $\mathcal{B}$ contains: β_{11} : $\Omega Acc(3, [6.8, 7.0))$, β_{12} : $\Omega Acc(2, [5.0, 5.2))$ and β_{13} : $\Omega Acc(1, [6.4, 6.6))$, and with a 10% decay in integrity for each time step: $\mathbf{P}(\beta_{11}) = 0.7$, $\mathbf{P}(\beta_{12}) = 0.2$ and $\mathbf{P}(\beta_{13}) = 0.9$

Eqn. 3 is used to calculate the distribution $\mathbf{W}_{\{\mathcal{K},\mathcal{B}\}}$ which shows that there are just five different probabilities in it. The probability matrix for the proposition $\Omega Acc(w, p)$ is:

$p \setminus w$	0	1	2	3	4
$[7.0, \infty)$	0.9967	0.9607	0.8428	0.7066	0.3533
$[6.8, 7.0)$	0.9803	0.9476	0.8330	**0.7000**	0.3500
$[6.6, 6.8)$	0.9533	0.9238	0.8125	0.6828	0.3414
$[6.4, 6.6)$	0.9262	**0.9000**	0.7920	0.6655	0.3328
$[6.2, 6.4)$	0.8249	0.8019	0.7074	0.5945	0.2972
$[6.0, 6.2)$	0.7235	0.7039	0.6228	0.5234	0.2617
$[5.8, 6.0)$	0.6222	0.6058	0.5383	0.4523	0.2262
$[5.6, 5.8)$	0.5208	0.5077	0.4537	0.3813	0.1906
$[5.4, 5.6)$	0.4195	0.4096	0.3691	0.3102	0.1551
$[5.2, 5.4)$	0.3181	0.3116	0.2846	0.2391	0.1196
$[5.0, 5.2)$	0.2168	0.2135	**0.2000**	0.1681	0.0840

In this array, the derived sentence probabilities for the three sentences in $\mathcal{B}$ are shown in bold type; they are exactly their given values.

Π's *negotiation strategy* is a function $\mathbf{S} : \mathcal{K} \times \mathcal{B} \rightarrow \mathcal{A}$ where $\mathcal{A}$ is the set of actions that send *Offer*(.), *Accept*(.), *Reject*(.) and *Quit*(.) messages to Ω. If Π sends *Offer*(.), *Accept*(.) or *Reject*(.) messages to Ω then she is giving Ω information about herself. In an infinite-horizon bargaining game where there is no incentive to trade now rather than later, a self-interested agent will "sit and wait", and do nothing except, perhaps, to ask for information. The well known bargaining response to an approach by an interested party "Well make me an offer" illustrates how a shrewd bargainer may behave in this situation.

An agent may be motivated to act for various reasons — three are mentioned. First, if there are costs involved in the bargaining process due *either* to changes in the value of the negotiation object with time *or* to the intrinsic cost of conducting the negotiation itself. Second, if there is a risk of breakdown caused by the opponent walking away from the bargaining table. Third, if the agent is concerned with establishing a sense of trust (Ramchurn et al., 2003) with the opponent —this could be the case in the establishment of a business relationship. Of these three reasons the last two are addressed here. The risk of breakdown may be reduced, and a sense of trust may be established, if the agent appears to its opponent to be "approaching the negotiation in an even-handed manner". One dimension of "appearing to be even-handed" is to be equitable with the value of information given to the opponent. Various bargaining strategies, both with and without breakdown, are described in (Debenham, 2004), but they do not address this issue. A bargaining strategy is described here that is founded on a principle of "equitable information gain". That is, Π attempts to respond to Ω's messages so that Ω's expected information gain similar to that which Π has received.

Π models Ω by observing her actions, and by representing beliefs about her future actions in the probability distribution $\mathbf{P}(\Omega Acc)$. Π measures the value of information that it receives from Ω by the change in the entropy of this distribution as a result of representing that information in $\mathbf{P}(\Omega Acc)$. More generally, Π measures the value of information received in a message, μ, by the change in the entropy in its entire representation, $\mathcal{J}_t = \mathcal{K}_t \cup \mathcal{B}_t$, as a result of the receipt of that message; this is denoted by: $\Delta_\mu|\mathcal{J}_t^\Pi|$, where $|\mathcal{J}_t^\Pi|$ denotes the value (as negative entropy) of Π's information in $\mathcal{J}$ at time t. Although both Π and Ω will build their models of each other using the same data, the observed information gain will depend on the way in which each agent has represented this information. To support its attempts to achieve "equitable information gain" Π assumes that Ω's reasoning apparatus mirrors its own, and so is able to estimate the change in Ω's entropy as a result of sending a message μ to Ω: $\Delta_\mu|\mathcal{J}_t^\Omega|$. Suppose that Π receives a message $\mu = \mathit{Offer}(.)$ from Ω and observes an information gain of $\Delta_\mu|\mathcal{J}_t^\Pi|$. Suppose that Π wishes to reject this offer by sending a counter-offer, $\mathit{Offer}(\delta)$, that will give Ω expected "equitable information gain". $\delta = \{\arg\max_\delta \mathbf{P}(\Pi Acc(\delta) \mid \mathcal{I}_t) \geq \alpha \mid (\Delta_{\mathit{Offer}(\delta)}|\mathcal{J}_t^\Omega| \approx \Delta_\mu|\mathcal{J}_t^\Pi|)\}$. That is Π chooses the most acceptable deal to herself that gives her opponent expected "equitable information gain" provided that there is such a deal. If there is not then Π chooses the best available compromise $\delta = \{\arg\max_\delta(\Delta_{\mathit{Offer}(\delta)}|\mathcal{J}_t^\Omega|) \mid \mathbf{P}(\Pi Acc(\delta) \mid \mathcal{I}_t) \geq \alpha\}$ provided there is such a deal. If there is not then Π does nothing.

6 CONCLUSION

Emergent processes are business processes whose execution is determined by the prior knowledge of the agents involved and by the knowledge that emerges during a process instance. The establishment of a sense of trust (Ramchurn et al., 2003) contributes to the establishment of business relationships and to preventing breakdown in one-off negotiation. The agent architecture is based on a first-order logic representation, and so is independent of the number of negotiation issues, although only two-issue bargaining is illustrated here. Emergent processes are managed by these agents that extract the process knowledge from the Internet using a suite of data mining bots.

REFERENCES

Bernhardt, D. and Miao, J. (2004). Informed trading when information becomes stale. *The Journal of Finance*, LIX(1).

Debenham, J. (2000). Supporting strategic process. In *Proceedings Fifth International Conference on The Practical Application of Intelligent Agents and Multi-Agents*, pages 237 – 256. Practical Applications Co.

Debenham, J. (2003). An eNegotiation Framework. In *The Twenty-Third International Conference on Innovative Techniques and Applications of Artificial Intelligence, AI'2003*, pages 79 – 92. Springer Verlag.

Debenham, J. (2004). Bargaining with information. In Jennings, N., Sierra, C., Sonenberg, L., and Tambe, M., editors, *Proceedings Third International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2004*, pages 664 – 671. ACM.

Dourish, P. (1998). Using metalevel techniques in a flexible toolkit for CSCW applications. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(2):109 – 155.

Fischer, L. (2003). *The Workflow Handbook 2003*. Future Strategies Inc.

Halpern, J. (2003). *Reasoning about Uncertainty*. MIT Press.

Jaynes, E. (1957). Information theory and statistical mechanics: Part I. *Physical Review*, 106:620 – 630.

Kraus, S. (2001). *Strategic Negotiation in Multiagent Environments*. MIT Press.

MacKay, D. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

Pietra, S. D., Pietra, V. D., and Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):380–393.

Ramchurn, S., Jennings, N., Sierra, C., and Godo, L. (2003). A computational trust model for multi-agent interactions based on confidence and reputation. In *Proceedings 5th Int. Workshop on Deception, Fraud and Trust in Agent Societies*.

Singh, M. (2004). Business Process Management: A Killer Ap for Agents? In Jennings, N., Sierra, C., Sonenberg, L., and Tambe, M., editors, *Proceedings Third International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2004*, pages 26 – 27. ACM.

Smith, H. and Fingar, P. (2003). *Business Process Management (BPM): The Third Wave*. Meghan-Kiffer Press.

van der Aalst, W. and van Hee, K. (2002). *Workflow Management: Models, Methods, and Systems*. The MIT Press.

ICEIS 2005

SEVENTH INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS

[illegible]eedings

Volume 4

MIAMI, U.S.A. · May 25-28, 2005

ORGANIZED BY

INSTICC
Institute for Systems and Technologies of Information, Control and Communication

CO-ORGANIZED BY

FIU
Florida International University
Miami's public research university

IN COOPERATION WITH

AAAI · acm SIGMIS

ICEIS 2005

Proceedings of the
Seventh International Conference on
Enterprise Information Systems

Volume IV

Miami, USA

May 25 – 28, 2005

Organized by
INSTICC – Institute for Systems and Technologies of Information, Control and Communication

Co-organized by
Florida International University

In cooperation with AAAI and ACM SIGMIS

Edited by Chi-Sheng Chen, Joaquim Filipe, Isabel Seruca and José Cordeiro

Printed in Portugal

ISBN: 972-8865-19-8

http://www.iceis.org

secretariat@iceis.org