Relating Quantified Motivations for Organizationally Situated Agents*

Thomas Wagner and Victor Lesser

Department of Computer Science
University of Massachusetts, Amherst, MA 01003
wagner@cs.umass.edu / (413) 545-3444

Abstract. To scale agent technologies for widespread use in open systems, agents must have an understanding of the organizational context in which they operate. Our current focus is on the expansion of agent knowledge structures to support modeling of organizational information and on a corresponding expansion of agent control techniques to use the information. In this paper we focus on the issue of task valuation and action selection in such socially situated agents. Specifically on the issue of quantifying agent relationships and relating work motivated by different sources. For example, the comparison of work done for self-interested reasons to work motivated by cooperative strategies.

1 Introduction

We believe that in order to scale-up agent technology [14] for use in open application domains, e.g., electronic commerce on the web, agents must model their organizational relationships with other agents and reason about the value or utility of interacting and coordinating with particular agents over particular actions. For example, a database management agent owned and operated by IBM[1] might have an extremely cooperative relationship with an information gathering agent owned by Lotus (Lotus is a subsidiary of IBM), but an entirely different type of relationship with a Microsoft information gathering agent – the IBM agent might prefer to service requests for the Lotus agent over the Microsoft agent or it might be willing to cooperate with the Microsoft agent if a higher fee is paid for its services. The agents might even coordinate via different protocols; the IBM agent might haggle with the Microsoft agent over delivery time and price whereas it might simply satisfy the Lotus request in short order and with a nominal or zero profit margin. Representing situations such as these is one aspect of our current research agenda. The overall objective is to expand the contextual information used by agents to make control decisions. Space limitations preclude a full description of the modeling or knowledge structures under consideration, however, the structures specify, or partially specify factors such as: the (multiple) organizations to which an

* Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Air Force Materiel Command, USAF, under agreement number F30602-97-1-0249 and by the National Science Foundation under Grant number IIS-9812755 and number IRI-9523419. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Air Force Research Laboratory, National Science Foundation, or the U.S. Government.

[1] Disclaimer: The use of actual corporate entities in this example is for illustration purposes only.

agent belongs, the different organizational roles an agent is likely to perform for the organization (in a task-centered sense, akin to [12]), the relationships between agents within the organization and without, the importance of a given role to an organization, the importance of a role to the agent, the coordination protocols to use in different circumstances, etc.

Broadening the scope of an agent's understanding of the organizational context in which it operates affects the agent control equation in two primary ways. *Structural* information affects the *scope* of the agent control process. For example, information that specifies with which agents a given agent is likely to interact, with respect to a particular goal, affects the scope of the agent's coordination dialogue. Structural information is particularly important in large MAS because it helps control the combinatorics – it may constrain the distributed search space for any coordination episode. In contrast, *value* information pertains mainly to representing, and reasoning about, complex agent relationships. Value information affects the way in which a given agent evaluates its candidate tasks and actions; information that describes the objective function [22] of an organization, and thus the relative importance of tasks performed for the organization, falls into this category. This characterization may appear over-simplified as interactions between coordination actions and problem solving actions may result in scope issues affecting value and vice versa. We differentiate between the types of information in this fashion because structural information pertains mainly to coordination activities whereas value information pertains mainly to the value or utility associated with particular problem solving options (including coordination actions). In this paper, we focus on the *value* side of the problem, i.e., on the agent's *in context* task valuation and selection process.

To ground the discussion, consider a simple example. Figure 1 shows an organized network of financial information agents in the WARREN [9] style. The network is a subset of a larger organization of agents that is populated by three types of agents. *Database Manager* (DBM) agents are experts in data maintenance and organization. These agents maintain repositories of information, e.g., D&B reports, Value Line reports, financial news, etc., and act as the interface between a repository or digital library and the rest of the network. The repositories may be simple databases, collections of databases, or even entail lower-level database management agents with which the primary database manager interacts. Thus the manager's functions are not simply to query a single existing database, instead they conform to the properties of agency, having multiple goals, multiple ways to achieve the goals, and so forth. *Information Gathering* (IG) agents are experts in particular domains. They know about databases (and database managers) pertaining to their area of expertise, or know how to locate such databases. Their task is to plan, gather information, assimilate it, and produce a report, possibly accompanied by a recommendation to the client about a particular action to take based on the gathered information. The IG agents pictured in the figure are both experts at collecting and assimilating financial news to build investment profiles of different companies. *Personal Agents* (PA) interface directly with the human client, perhaps modeling the client's needs. These agents also decide with which information specialists to interact to solve a client's information need. PAs for a given company may interact with specialists outside of the company, however, interaction styles may

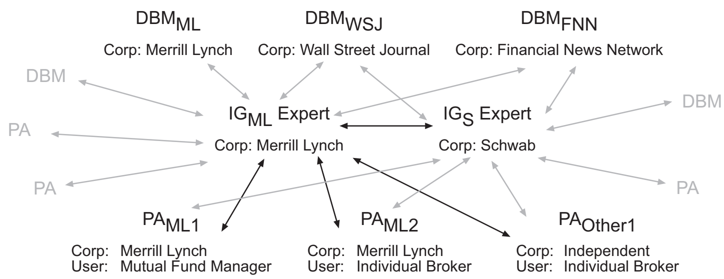
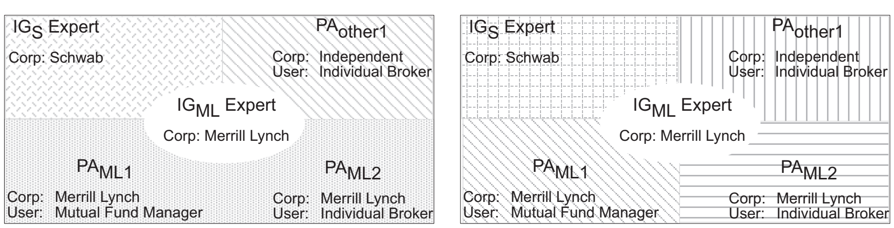


Fig. 1. A Network of Organized Information Agents

differ, i.e., different protocols may be used, different fee structures may apply, etc. The edges in the figure denote interactions between agents. We will focus on the interactions and relationships between the IG expert for Merrill Lynch, denoted IG_{ML}, and the IG expert for Schwab, and the multiple pictured PAs.

Agent IG_{ML} is organizationally situated. The agent belongs to multiple different organizations and it has different relationships with the other agents, stemming from the different organizations, different organizational objectives within and without the organizations [3], and from different relationships within the organizations. Figure 2(a) shows IG_{ML}'s organizational relationships. It is part of the Merrill Lynch corporate structure and thus shares this organization with PA_{ML1} and PA_{ML2}. It is also part of the set of IG agents that specialize in financial information gathering and shares this in common with IG_S. IG_{ML} also belongs to the organization of financial information agents and shares this in common with PA_{Other}. Note, we view organizations as hierarchical structures that can be specialized (i.e., subclassed). In this figure, the organization shared by IG_{ML} and PA_{Other} may have the same root as the organization shared by IG_{ML} and IG_S, however, the specializations differ (in fact, all the agents are members of a root organization pertaining to financial information agents). In addition to its organizational positioning, IG_{ML} also has different relationships within these organizations. Figure 2(b) shows the agent's different relationships. This figure differs from Figure 2(a) in that IG_{ML} has a different relationship with PA_{ML1} and PA_{ML2}. While PA_{ML1} and PA_{ML2} are both members of the Merrill Lynch organization, PA_{ML1} represents a mutual fund manager from the funds division and PA_{ML2} represets an individual broker associated with the retail division.

One of the issues that arises when examining a scenario like this is the need to relate the different motivational factors that influence agent decision making. For example,



(a) IG_{ML}'s Organizational Memberships (b) IG_{ML}'s Inter-agent Relationships

Fig. 2. Different Relationships Complicate Action Choice

IG_{ML} interacts with PA_{ML1} and PA_{ML2} for cooperative reasons. In contrast, IG_{ML} interacts with PA_{Other} for self-interested reasons, namely profit for itself, its division, or Merrill Lynch. Agents situated in open, social, environments interact with different agents, and different organizations of agents, for different reasons. The ability to relate these different motivations is a requisite for the agents to act rationally, or approximately so, given their social context. Without this ability, how can IG_{ML} determine which requests to service, and in what order? Assuming a model in which agents are rationally bounded, tasks/requests arrive dynamically, and deadlines or service times on requests are present, the agent cannot simply perform all the tasks, but must instead select a subset of the tasks to perform and then determine an appropriate sequence in which to perform them. It is important to note that the agent decision process is contextual. Since the environment is dynamic, and the state of problem solving changes over time, given a set of tasks from which to choose, the choice of which tasks are appropriate is dependent on the current situation. For instance, if IG_{ML} has spent the last n units of time problem solving for PA_{ML1}, and new requests from PA_{ML1} and PA_{ML2} arrive, even if PA_{ML1} requests generally take precedence over PA_{ML2} requests (as specified by the organizational structure), it may be appropriate for IG_{ML} to service the PA_{ML2} request before servicing the PA_{ML1} request.

Figure 3 shows IG_{ML}'s candidate actions at some point time, t. The tasks are structure in a TÆMS [10] network, though the *sum()* function simply specifies that any number of the tasks may be performed in any order. IG_{ML}'s candidate tasks include servicing requests from PA_{ML1}, PA_{ML2}, and PA_{Other}, as well as doing a local-only task (updating its source models). It also has the option of contracting out its *update sources* task to IG_S. In order to compare these actions the agent requires a framework that quantifies and relates the different motivational reasons for performing particular tasks, as well as relating the costs/benefits of doing tasks for others, and doing local work, to the costs/benefits associated with contracting out the local update task. The complexity of the relationships that the agent has with other agents requires this complex approach to evaluation. The rationale for keeping the different motivational concerns separate is that they represent quantities that are not interchangeable, e.g., progress toward different problem solving objectives, akin to [18]. They are not reducible at all agents to some uniform currency and not all quantities have value to all other agents. For example, doing a favor for someone cannot in turn be used to purchase something at the local store. Another intuitive example: work done on one's yard has no intrinsic value to a professional peer, unless said peer is your neighbor. With respect to computational agents, partitioning of concerns like these maps to the balancing of local work with non-local work, but also to the balancing of work done to satisfy some joint goal JG_α in con-

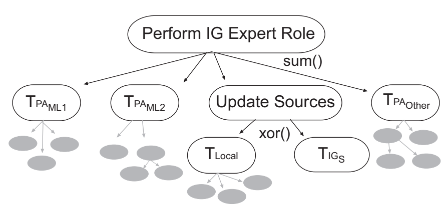


Fig. 3. IG_{ML}'s Abstracted Task Structure

trast to work done to satisfy joint goal JG_β. The idea of this research is not wholly to partition different activities, and the evaluation of their *worth* to the agent, but rather to support ranges of representations, e.g., tasks and actions that have both self-interested and cooperative motivations, or work relating to multiple different joint goals held by multiple agents related, at least partially, through different organizations.

In the sections that follow we present a model for relating different motivational factors, and different measures of progress, that enables agents to compare different types of actions, and the costs and benefits of particular courses of action. We then discuss the issue of interfacing this model with our existing agent control technologies and present ideas about how agents will make decisions based on this model.

2 Quantifying and Comparing Motivations

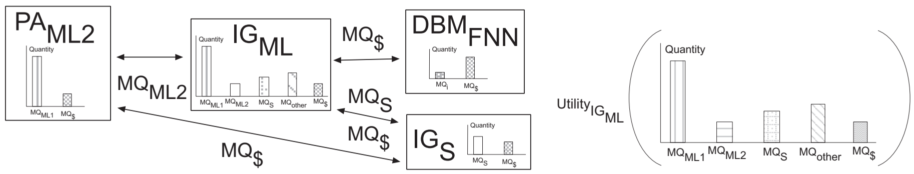
There are three different classes of tasks that a socially situated agent, such as IG_{ML}, must reason about: 1) tasks that are of local concern only and do not have direct value or repercussions in any non-local context; 2) tasks that other agents wish the local agent to perform; and 3) tasks that other agents may perform for the local agent. Obviously, there are graduations or tasks that pertain to more than one of these classes. For example, a task may produce a result that is valuable locally as well as having value to another agent. Additionally, each task may be performed for cooperative reasons, self-interested reasons, or ranges of these. For example, performing a task for an associate for a nominal fee may pertain to both cooperative concerns and self-interested motivations. It is important to note that even actions performed for cooperative reasons actually have different motivations. For example, doing a favor for one's superior at work is evaluated differently than doing a favor for a peer, which is treated differently than doing a favor for persons unknown, and so forth. In order to address these concerns, we have developed a model for agent activities that quantifies these different motivational factors and enables the local agent to compare the factors via a multi-attributed utility function. Definitions:

Agents are autonomous, persistent, computing entities that have the ability to choose which tasks on which to operate, and when to perform them. Agents:

- Can perform tasks locally if they have sufficient resources.
- Interact with other agents to perform tasks. This entails the local agent asking other agents to perform tasks, or the local agent performing tasks for other agents.[2]
- Agents interact via multiple different mediums of exchange known as *motivational quantities* (MQs) that are produced by performing tasks, i.e., a given agent has a set of MQs that it accumulates and exchanges with other agents, as shown in Figure 4(a).[3]

[2] This model subsumes results sharing and coordination through side effects. If an agent has already produced a result that another agent needs, then the other agent's task has already been performed. The existence of the result at hand may affect the "price" charged for the results, but, it does not affect the abstract modeling approach described here.

[3] If agents are allowed to contract with other agents via a proxy agent, and the proxy agent translates MQs of one type to another, it is possible for the agents to be viewed as sharing a common MQ. However, this is limited by the availability of MQs of the proper type. If we ignore the issue of MQ quantity, the general issue of reducibility of MQs via proxy can be viewed as a graph connectivity problem.



(a) Agents Interact via Different MQs (b) Utility is Based on MQ State

Fig. 4. Role of MQ in Agent Control

- Not all agents have the same MQ set. However, for two agents to interact, they must have at least one MQ in common (or have the means for forming an MQ dynamically).
- For each MQ_i belonging to an agent, it has a preference function or utility curve[4], U_{f_i}, that describes its preference for a particular quantity of the MQ, i.e., $\forall MQ_i,\ \exists U_{f_i}()$ such that $U_{f_i}(MQ_i) \ \mapsto\ U_i$ where U_i is not directly interchangeable with U_j unless $i\ =\ j$. Different agents may have different preferences for the same MQ_i.
- An agent's overall utility at any given moment in time is a function of its different utilities: $U_{agent}\ =\ \gamma(U_i,\ U_j,\ U_k,\ ..)$, as shown in Figure 4(b). We make no assumptions about the properties of $\gamma()$, only that it enables agents to determine preference or dominance between two different agent states with respect to MQs.
- For simplicity of presentation, let us assume that $\gamma()$ is not a multi-variate utility function and instead that for that for each U_i there is an associated function $\omega_i()$ [5] that translates MQ specific utility into the agent's general utility type, i.e., $\forall U_i,\ \exists \omega_i()$ such that $\omega_i(U_i)\ \mapsto\ U_{agent}$. Thus U_{agent} may take the form of Equation 1.[6]

$$U_{agent} = \sum_{i=0}^{n} \omega_i(U_i)\ \ and\ \ \Delta U_{agent}\ =\ |\sum_{i=0}^{n} \omega_i(U_i') - \omega_i(U_i)| \qquad (1)$$

Tasks are abstractions of the primitive actions that the agent may carry out. We return to the issue of abstraction in Section 4. Tasks:

- Require some time or duration to execute, denoted d_i.
- May have deadlines, $deadline_i$, for task performance beyond which performance of said task yields no useful results. (This could also be defined via a function that describes a gradual decrease in utility.)
- Produce some quantity of one or more MQs, called an *MQ production set* ($MQPS$), and denoted: $MQPS_{i,j,k}\ =\ \{q_i, q_j, q_k, ..\}$, where $\forall i,\ q_i \geq 0$. These quantities

[4] We currently view these as continuous functions but are exploring the possible need for step-wise utility functions that describe "saving-up" for a potential future event.

[5] Astute readers will note that $\omega_i()$ could be combined with $U_{f_i}()$. We partition these concerns to provide separate places for mapping different organizational and relationship-centered influences.

[6] This simple model assumes that all utilities associated with different motivational quantities can be mapped to some common denominator at the agent. This does not mean that the same mapping is possible at all agents, nor do we feel this property is necessary for the model. It is intended to simplify presentation and model manipulation at this time.

are *positive* and reflect the benefit of derived from performing the task. They may be the direct outcome of performing the task, i.e., some result produced by doing the actual work, or they may be quantities that another agent is paying for the work to be performed. In this model, the two are equivalent.

– Tasks may have multiple MQ production sets; that is a given task may produce different groups of MQs. This models the idea that agents may interact with multiple different mediums-of-exchange. For instance, agent IG_{ML} may service a request for agent IG_S in return for some financial compensation, or by IG_S "calling-in" a favor, or for some combination of these. The multiple MQ production sets are represented: $\{MQPS_{i,j},\ MQPS_{l,m},\ ..\} = \{\ \{q_i, q_j\},\ \{q_l, q_m\},\ ..\}$. Note that $MQ_x \cap MQ_y$ may $\neq\ \phi$ as different $MQPS$ sets may have common members. To simplify presentation, we concentrate on tasks that have a single $MQPS$, though we return to the issue of different $MQPS$ in Section 3.[7]
– Akin to the $MQPS$, tasks may also consume quantities of MQs. The specification of the MQs consumed by a task is called an ***MQ consumption set*** and denoted $MQCS_{i,j,k} = \{q_i, q_j, q_k, ..\}$, where $\forall i,\ q_i \geq 0$. As with $MQPS$s, a task may have multiple $MQCS$ sets. Consumption sets model the idea of tasks consuming resources and agents contracting work out to other agents, e.g., paying another agent to produce some desired result or another agent accumulating favors or good will as the result of task performance. In contrast to production sets, consumption sets are the *negative* side of performing a particular task.
– All quantities, e.g., d_i, $MQPS$, $MQCS$, are viewed from an expected value standpoint. We return to the issue of uncertainty in Section 5.

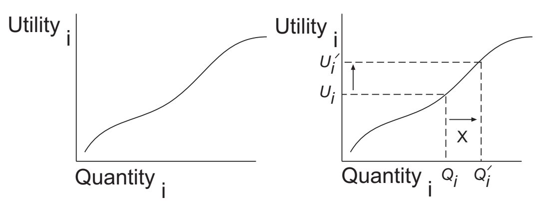


Fig. 5. Motivational Quantities and Utility

To illustrate, Figure 5 shows a single utility curve for a single MQ. Assume some task, T, produces X amount of MQ_i. The agent reasons about task performance, and the utility thereof, by comparing the change in U_i associated with the change in MQ_i that performing T will produce. If this is the only task being considered, $\Delta U_{agent} = \Delta U_i$.

Figure 6 illustrates the model's application to the task structure of IG_{ML} pictured in Figure 3. The different problem solving options available to IG_{ML} are: 1) performing task $T_{PA_{ML2}}$ for PA_{ML2}; 2) performing task $T_{PA_{Other}}$ for PA_{Other}; 3) performing its

[7] The issue of which $MQPS$ from the candidate sets will pertain to a given transaction can be viewed as an issue for explicit negotiation between agents [13], or as a dynamic agent choice problem [23].

local update task, T_{Local}; 4) contracting its local update task out to IG_S, represented as T_{IG_S}. Recall that IG_{ML} has different relationships with PA_{ML2}, PA_{Other}, and IG_S. As shown in Figure 4(a), the agents' different relationships translate into different MQs with which they interact. IG_{ML} services requests from $PA_{ML2}$for cooperative reasons – it is part of IG_{ML}'s job description and it is recorded as an inter-company transaction for reporting purposes. This motivation is expressed as MQ_{ML2} in IG_{ML}'s $MQPS$. In contrast, IG_{ML} has a very different relationship with IG_S– per the two agents' $MQPS$ sets, they may interact via currency ($MQ_\$$) or via an MQ based on professional favors, classified as MQ_S. IG_{ML} has still another relationship with PA_{Other} and they interact via currency only. To compare the different candidate tasks, IG_{ML} reasons about the positive/negative changes in utility that result from carrying out the tasks. For example, to compare T_{Local}, $T_{PA_{ML2}}$, and $T_{PA_{Other}}$ (assuming the single valued utility mapping shown in Equation 1):

1. For $T_{PA_{ML2}}$: 1) The task consumes a local resource U_{R1}, e.g., monthly allotment of ppp connection time. Therefore, compute the negative change in U_{R1} that will result from the performance of $T_{PA_{ML2}}$; 2) compute the positive change in U_{ML2} that is produced by performing the task for PA_{ML2} (i.e., the increase in MQ_{ML2}); 3) $U'_{agent_{scenario:ML2}} = \omega(U'_{R1}) + \omega(U'_{ML2})$.
2. For $T_{PA_{Other}}$: 1) compute the negative change in U_{R2}, another (different) local resource that is consumed by $T_{PA_{Other}}$; 2) compute the positive change in $U_\$$ that is produced by performing the task for PA_{Other} (i.e., the monetary payment from PA_{Other} to IG_{ML}); 3) $U'_{agent_{scenario:PA_{Other}}} = \omega(U'_{R2}) + \omega(U'_\$)$.
3. For T_{Local}: 1) compute the positive change in U_L produced by the performance of task T_{Local}; 2) $U'_{agent_{scenario:Local}} = \omega(U'_L)$.
4. To select between the three, simply choose whichever has the highest gain in utility for the agent. For example, if $U_{agent_{scenario:Local}} \geq U_{agent_{scenario:PA_{Other}}}$ and $U_{agent_{scenario:Local}} \geq agent_{scenario:ML2}$ then perform the local action. In other words, if the gain in utility achieved by performing T_{Local} exceeds the utility produced by performing $T_{PA_{ML2}}$, even when considering the resource cost of $T_{PA_{ML2}}$ (note that U'_{R1} is less than U_{R1} in Figure 6), then it is preferable to perform T_{Local}. Likewise with $T_{PA_{Other}}$.

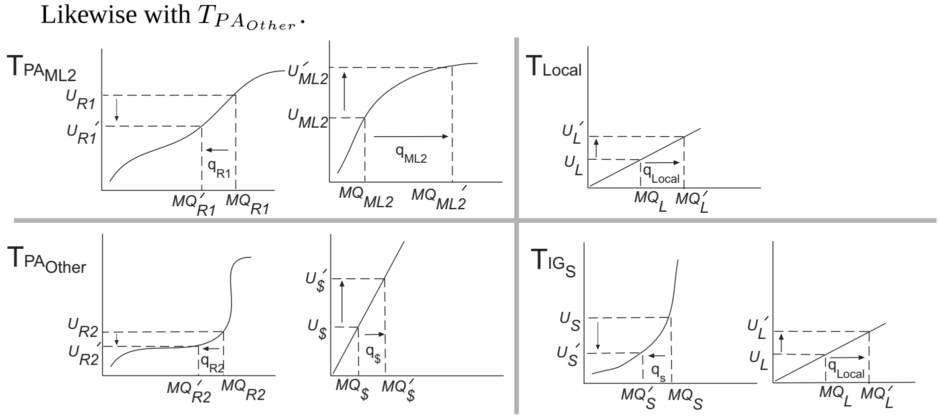


Fig. 6. Comparing Different Candidate Tasks

If the agent's objective is to simply select which task to perform next, and tasks do not have associated deadlines, and the present and future value of MQs are equivalent, then it can reason using the maximum expected utility principle and select the task at each point that maximizes immediate utility. However, this simple choose-between-available-tasks model does not map well to situations in which tasks have deadlines, or even situations with a temporal component. For example, consider choosing between T_{Local} and T_{IG_S}: if $\omega(U'_L) \geq \omega(U'_S) + \omega(U'_L)$ then perform the task locally, otherwise, contract it out. In this case, $\omega(U'_S)$, which is the cost of having IG_S perform the task for IG_{ML}, must be zero in order for IG_{ML} to consider allocating the task to IG_S. In order to properly assess the value of such an arrangement, the agents need to use the model presented in this section for comparisons, but, to add components such as *opportunity cost* or *future value* to the selection / decision process. We return to this issue in Section 5.

In this section we have presented a model for comparing tasks that are motivated by different factors. The model can support comparison between tasks that are performed for other agents in return for financial gain to tasks that are performed for other agents for cooperative reasons. Via the different preferences for the different quantities, agent control can be modulated and agents can reason about mixtures of different task types and different motivations. For example, a socially situated agent can reason about doing work in exchange for money as well as the accumulation of goodwill, favors, and other non-currency exchanges. The use of state in the model also facilitates contextually dependent behaviors or adjustments to behaviors over time. Agent α performing cooperative work with a closely allied agent, β, for instance, may need to balance this work with cooperative work with others over time. As α accumulates goodwill (represented as one MQ) with β, its preference may shift to the accumulation of other MQs. The use of utility for this application is flexible and very general, though to effectively use the model we must address how to meaningfully plan and reason with the model and how to integrate it into existing agent control technology. We return to these issues in Sections 4 and 5.

3 Incorporating Organizational Structure and Influence

The MQ model enables the direct comparison for work motivated by variety of different sources, and it supports ranges of these. However, the model also supports the integration of certain classes of organizationally derived influence and structure. For instance, organizational relationships can be associated with particular MQs, i.e., agents belonging to a particular organization and interacting for a particular organizational goal, can use an MQ explicitly for that purpose. Thus agents belonging to the organization can reason about their contributions to group problem solving and the contributions of others. Additionally, the same agents may then belong to different organizations, each having its own MQ – the model enables each agent to compare and assess its contributions to multiple different concerns.

The selection of different $MQPS$ and $MQCS$ is another place where organizational structure integrates with the model. Organizations may have relationships with each other and this can be mapped into the selection of MQs in particular $MQPS$/$MQCS$ sets. For instance, if organization α related to organization β in such a way that members of α are willing to coordinate in a cooperative fashion, though to a limited extent, with members of β, agents belonging to α can exchange MQ_{α} as well as $MQ_{\alpha\beta}$. The notion of "limited extent" in the previous sentence points to another place where organizational structure maps into the MQ-centric model; the preference functions or utility curves of the agent reflect the relative importance of particular types of problem solving activities to the agent. For example, a type of problem solving that is very important to the agent will have a steep utility curve relative to its other concerns. If the importance is uniform, the utility curve will have a constant (linear) slope; this approach also pertains to power relationships between agents. Organizational influences and relationships can also be mapped to γ, or to the ω functions used in the utility mapping of Equation 1.

Organizational structure imposed on the computation also comes into play in the initial assignment of q_i's (quantities of MQs) to agents. Note that since work is produced over time, the system is not a *zero sum game*, but instead is a growing economy. However, the initial allocation of MQs to agents predisposes the system to initialize in a particular way and biases the flow of the distributed computation, as in [18].

Agent communication and the use of default knowledge also have roles in this model. Negotiation [13] between agents can be used to select which MQs, from a set of candidate $MQPS$ / $MQCS$, will be used for a given transaction. Negotiation can also be used to determine the "price" (in MQs) or quantity that a particular transaction will produce. Auctions or other market mechanisms [27, 6, 4] can be integrated with the model through this avenue. Space limitations preclude meaningful discussion of the organizational influences into the model, but we feel that the ability to support organizational structure in an agent's contextual decision / evaluation process is an important asset of this approach.

4 Integration with Detailed Agent Control

The MQ model is deliberately abstract to simplify control reasoning at the *meso*-level of agent control [19], i.e., the computational organizational level rather than the micro-level. While it could be used directly at the micro-level of agent control, the agent would be unable to reason about a wide class of issues that are important for socially situated, resource bounded, agents. The model lacks features such as explicit representation and quantification of interactions[8] between tasks and a detailed view of the actions that may be used to carry out the tasks. We generally subscribe to a model where agents have alternative ways to perform tasks (or achieve goals), and that part of the agent control problem is to evaluate the different possible ways to perform a task, taking into consideration the different trade-offs or performance characteristics, and to select one or more from the set of alternatives. Additionally, detailed and complex interactions between agent activities, such as chains of interactions, motivate detailed coordination between agents. This detailed, quantitative, temporal, constraint and interaction based view of the world is embodied by research in TÆMS [11], Design-to-Criteria (DTC) agent scheduling [25], and GPGP [10] agent coordination.

The existence of such sophisticated, quantitative, machinery for agent control begs the question of why the MQ-centered model is necessary. The detailed technologies

[8] However, we are considering certain classes of interaction modeling at this level; the issue is expressiveness versus tractability.

are well suited to representation and control at a particular level of detail (micro-level). However, the representational power of the TÆMS modeling language has a downside – complexity. Both GPGP and DTC cope with the combinatorics in different ways, however, expanding the TÆMS model further, both in terms of detail and in terms of problem solving scope, is undesirable. Through the MQ-model we aim to support a new class of reasoning about the agent's social environment, its organizational context, while not adding to the complexity of DTC, GPGP, or the TÆMS modeling language itself.

The integration of the MQ world view with the detailed tools is akin to other recent work in integrating process-program controllers [15, 28] and opening the detailed tools for use with BDI problem solvers [5, 20] and others [26, 16]. The general view is that other high-level agent control components, or even high-level agent coordinators, are responsible for influencing the selection of candidate tasks for the agent. In other words, the responsibilities are partitioned: at one level components like BDI problem solvers or organizational context experts (using the MQ model) reason about high-level task and goal selection, possibly by exchanging information with other agents. At another level, GPGP/DTC/TÆMS is used to perform feasibility analysis, to evaluate the detailed temporal and resource constraints of the different tasks (and the different primitive actions), and to form commitments between agents to sequence activities over interactions and so forth. On one hand, abstract reasoning about tasks where optimality or domain specificity are ideas to consider, and on the other hand, satisficing, real-time, detailed, temporal control or *implementation* of the selected tasks and goals. In terms of the MQ model itself – a reasoner using the model can work with other high-level components to select the set of candidate tasks and goals for the agent, as well as modulate the lower-level feasibility and "implementation" tools by mapping MQ preferences into the quality, cost, and duration used by these tools to reason about candidate tasks. Space precludes a detailed discussion, but, the integration takes place on multiple fronts and also requires a two-way interface between the high-level and low-level controllers. Intuitively, as the high-level controllers lack the detailed, temporal, view, it is possible to select candidate tasks that are unachievable (unimplementable), or unachievable in any desirable way (per goal criteria) [24].

5 Future Directions

The model presented here is currently under development and integration. However, the model stands on its own merits as a way to quantify and relate hereto unrelated concerns like cooperative and self-interested motivational factors. Using the model, agents can reason about different concerns like self-interest, favors, altruism and social welfare [8][9]. The model also frames the problem of balancing these different motivations, as well as balancing work between different organizational entities and balancing different agent relationships. The model relates to other recent work in the multi-agent community, such as agents interacting via obligations [2], or notions of social commitment [7],

[9] All mapped to different MQs or groups of MQs. However, the issue of how to specify system-wide goal criteria, or organizational-level goals, that characterize acceptable ranges of these must also be addressed to employ MQs to concepts like social welfare in a meaningful fashion.

but it differs in its quantification of different concerns and its dynamic, contextual, relative, evaluation of these. The model resembles MarCon [18] as the different degrees-of-satisfaction afforded by the MQ model is related to MarCon's constraint optimization approach, and MarCon too deals with utilities/motivations that cannot always be commingled. MarCon, however, views constraints as agents, assigning particular roles to particular agents, and the issue of which tasks to perform do not enter into the problem space.

With respect to the work presented here, many research questions remain. Aside from the obvious (and deliberate) lack of prescriptive semantics for the model, the questions that remain include how to best leverage the model from a decision making standpoint, i.e., how to incorporate the model into a high-level decision process that can then be integrated with the rest of our agent control technology as discussed in Section 4. The decision processes currently being considered include facets that factor-in the future value of MQs, durations and deadlines, and the opportunity cost of choosing one task over another. The importance of the future value question is best illustrated in the context of cooperative coordination research, e.g., tit-for-tat agent coordination [21] and other cooperative games [17]. If a cooperative agent has reason to believe that future requests sent to other cooperative agents will not be carried out, then it has little or less incentive to service requests for the other agents at the current time (reciprocity). In the MQ view of the world, this would entail applying a present value to the MQs and their associated utility that reflects the belief in the situation that will arise downstream temporally, e.g., if all cooperation with another agent is likely to cease, it makes little sense to accumulate MQs for use with said agent. Opportunity cost is important for related reasons – the selection of one task may preclude performance of another as tasks may have deadlines. Reasoning about decommitment penalties or costs [1] also factors into the model at this level; if an agent commits and then decommits, the opportunity cost of the task it has chosen instead may include the penalty in MQs to the agent for its decommitment. Other research questions pertain to the role of uncertainty in the MQ model and the integration of MQ-based experts with other high-level agent control components.

References

1. Martin Andersson and Tumas Sandholm. Leveled commitment contracts with myopic and strategic agents. In *Proc. of the 15th National Conf. on AI (AAAI)*, pages 38–44, 1998.
2. Mihai Barbuceanu. Agents that work in harmony by knowing and fulfiling their obligations. In *Proc. of 15th National Conf. on AI (AAAI)*, pages 89–96, 1998.
3. Jonathan Boswell. *Social and Business Enterprises*. George Allen & Unwin Ltd., London, England, 1976.
4. Craig Boutlier, Yoav Shoham, and Michael P. Wellman. Economic Principles of Multi-Agent Systems. *Artificial Intelligence*, 1-2(1-6), 1997.
5. M.E. Bratman. *Intention Plans and Practical Reason*. Harvard University Press, Cambridge, MA, 1987.
6. Donald E. Campbell. *Resource Allocation Mechanisms*. Cambridge University Press, 1987.
7. Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In *Proc. of the 1st Intl. Conf. on Multi-Agent Systems (ICMAS)*, pages 41–48, 1995.
8. James Coleman. *Individual Interests and Collective Action*. Cambridge University Press, Cambridge, England, 1986.

9. K. Decker, A. Pannu, K. Sycara, and M. Williamson. Designing behaviors for information agents. In *Proc. of the 1st Intl. Conf. on Autonomous Agents*, pages 404–413, February 1997.
10. Keith S. Decker. *Environment Centered Analysis and Design of Coordination Mechanisms*. PhD thesis, University of Massachusetts, 1995.
11. Keith S. Decker. Task environment centered simulation. In M. Prietula, K. Carley, and L. Gasser, editors, *Simulating Organizations: Computational Models of Institutions and Groups*. AAAI Press/MIT Press, 1996.
12. Edmund H. Durfee and Thomas A. Montgomery. Coordination as distributed search in a hierarchical behavior space. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1363–1378, 1991.
13. P. Faratin, C. Sierra, and N. Jennings. Negotiation Decision Functions for Autonomous Agents. *Intl. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1997.
14. Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):8–38, 1998.
15. David Jensen, Yulin Dong, Barbara Staudt Lerner, Eric K. McCall, Leon J. Osterweil, Stanley M. Sutton Jr., and Alexander Wise. Coordinating agent activities in knowledge discovery processes. In *Proc. of Work Activities Coordination and Collaboration Conf. (WACC)*, 1999.
16. Victor Lesser, Bryan Horling, Frank Klassner, Anita Raja, Thomas Wagner, and Shelley XQ. Zhang. BIG: A resource-bounded information gathering agent. In *Proc. of the 15th National Conf. on AI (AAAI)*, July 1998.
17. Martin A. Nowak, Robert M. May, and Karl Sigmund. The Arithmetics of Mutual Help. *Scientific American*, pages 76–81, June 1995.
18. H. Van Dyke Parunak, Allen Ward, and John Sauter. A Systematic Market Approach to Distributed Constraint Problems. In *Proc. of the 3rd Intl. Conf. on Multi-Agent Systems (ICMAS)*, 1998.
19. Michael J. Prietula, Kathleen M. Carley, and Les Gasser. A Computational Approach to Oganizations and Organizing. In Michael J. Prietula, Kathleen M. Carley, and Les Gasser, editors, *Simulating Organizations: Computational Models of Institutions and Groups*, pages xiv–xix. AAAI Press / MIT Press, 1998.
20. A.S. Rao and M.P. Georgeff. Modelling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proc. of the 3rd Intl. Conf. on Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann, 1991.
21. Sandip Sen and Anish Biswas. Effects of misconception on reciprocative agents. In *Proc. of the 2nd Intl. Conf. on Autonomous Agents (Agents)*, pages 430–435, 1998.
22. Herbert Simon. Organizational objective functions. Personal Conversation, 1997.
23. Ludwig von Auer. *Dynamic Preferences, Choice Mechanisms, and Welfare*. Springer-Verlag, Heidelberg, Germany, 1998. Chapter 8.
24. Thomas Wagner, Alan Garvey, and Victor Lesser. Complex Goal Criteria and Its Application in Design-to-Criteria Scheduling. In *Proc. of the 14th National Conf. on AI (AAAI)*, pages 294–301, July 1997.
25. Thomas Wagner, Alan Garvey, and Victor Lesser. Criteria-Directed Heuristic Task Scheduling. *Intl. Journal of Approximate Reasoning, Special Issue on Scheduling*, 19(1-2):91–118, 1998. See also UMASS CS TR-97-59.
26. Thomas Wagner and Victor Lesser. Toward Generalized Organizationally Contexted Agent Control. In *AAAI Workshop on Reasoning in Context*, 1999.
27. Michael Wellman and Peter Wurman. Market-Aware Agents for a Multiagent World. In *Proc. of the 2nd Intl. Conf. on Autonomous Agents (Agents)*, pages 115–125, 1998.
28. Alexander Wise, Barbara Staudt Lerner, Eric K. McCall, Leon J. Osterweil, and Stanley M. Sutton Jr. Specifying Coordination in Processes Using Little-JIL. UMASS CS TR UM-CS-1998-038, 1998.