Applying Queuing and Probability Theory to Predict Organizational Behaviors [*]

Bryan Horling[†]
Multi-Agent Systems Lab
University of Massachusetts,
Amherst, MA
bhorling@cs.umass.edu

Victor Lesser
Multi-Agent Systems Lab
University of Massachusetts,
Amherst, MA
lesser@cs.umass.edu

ABSTRACT

Most existing organizational design processes focus on either the qualitative or domain-independent features of candidate designs. This paper demonstrates the significance of domain-specific features through an examination of an organizationally-driven information retrieval network. The behavior of a search process for appropriate agents and the consequences of hierarchical control in a continuous work flow are described. A model capable of predicting these and other characteristics is then created using techniques from queuing and probability theory. This model can then be used to guide the search for an appropriate design.

1. INTRODUCTION

Much of the existing work on organization design in multi-agent systems has focused on the qualitative aspects of those designs, or on a predefined set of numeric but largely domain-independent characteristics. In our recent work [4] we have developed a new representation called ODML that incorporates arbitrary quantitative information into the organizational model. Such a model can be used to make detailed predictions of how candidate designs will perform in different circumstances, which can be used as part of a larger search process through the space of design alternatives. Although such models can be more difficult to construct, we believe the benefits this more refined view can provide warrant research.

In particular, we believe that the ability to make concrete, numeric predictions about a range of domain-specific organizational characteristics can improve the quality of an organizational design process by capturing the complexities that exist in realistic systems. In contrast to solutions that use instrumented simulations to making similarly detailed predictions, such explicit, quantitative models also offer the possibility of very rapid design evaluation. This speed facilitates the search of the very large design space that agent organizations frequently have.

In this paper we will demonstrate this potential by describing how an organizationally-driven information retrieval system has been modeled in the ODML framework. Techniques from queuing and probability theory embedded in the model are used to capture several different characteristics, including a utility-driven search in the agent network and the effects of hierarchical control, both of which are common in multi-agent systems. These features are then joined in a common utility function that is used to provide a high-level comparative metric. The model itself has been validated through comparisons to empirical tests carried out in a simulation environment. The end product of the modeling process is a single, succinct construct that both describes the space of alternatives and predicts the performance of individual designs.

There are three principle contributions of this work. The first is an example of the flexibility and detail that are possible using the ODML framework. The second is a demonstration that it is possible to capture the relevant organizational characteristics of this domain, and integrate those features into a single predictive model. This demonstration also depicts the utility of the techniques from queuing and probability theory that we have used. Finally, this work also shows how complex, domain-specific system characteristics can interact with the organizational design to affect global performance. This fact motivates the detailed approach we take, and suggests that if such information is missing or abstracted away that the quality of a corresponding design process may suffer.

2. INFORMATION RETRIEVAL DOMAIN

A general peer-to-peer information retrieval (IR) system is composed of a number of interconnected databases, controlled by a set of (agent) entities. Queries are first received by individual members of the network. An appropriate set of information sources must then be discovered that can address the query, after which the query is routed and processed to produce a response for the user. The information necessary for responding to a particular query may be distributed across the network, which can cause an undirected retrieval process to be time consuming, costly, or ineffective, particularly when the number of sources is large.

The information retrieval model presented here is inspired by work by Zhang et al. [10], which proposes that a hierarchical organization can be used to address this problem. This solution organizes information sources in hierarchies, allowing queries to quickly prop-

[*] This material is based upon work supported in part by the National Science Foundation Engineering Research Centers Program under NSF Award No. EEC-0313747. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[†] Primary author is a student.

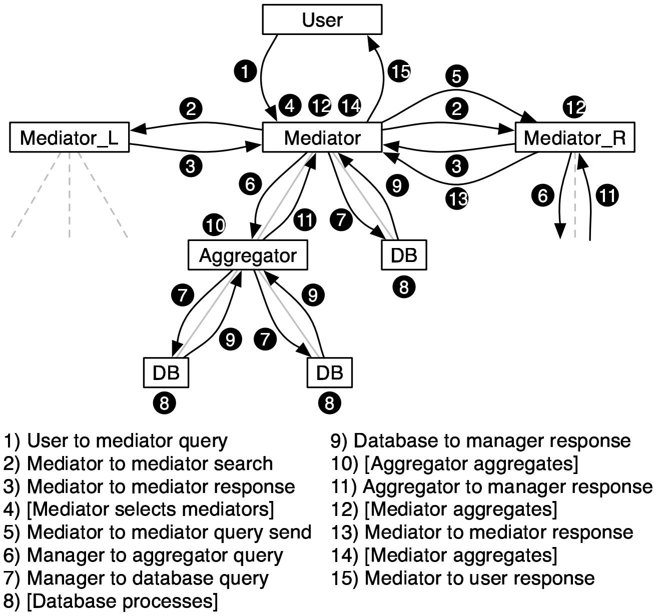

User
Mediator_L
Mediator
Mediator_R
Aggregator
DB
DB
DB
1) User to mediator query
2) Mediator to mediator search
3) Mediator to mediator response
4) [Mediator selects mediators]
5) Mediator to mediator query send
6) Manager to aggregator query
7) Manager to database query
8) [Database processes]
9) Database to manager response
10) [Aggregator aggregates]
11) Aggregator to manager response
12) [Mediator aggregates]
13) Mediator to mediator response
14) [Mediator aggregates]
15) Mediator to user response

Figure 1: The control and communication sequence involved in handling a query in the information retrieval organization.

agate to data sources, and results be routed back to a single agent in the network. At the top level of each hierarchy is a mediator. Each mediator is responsible for providing a concise and accurate description, known as a collection signature, of the data available in the information sources present in the hierarchy below it. An information source may be an individual database, or an aggregator which manages other sources. Mediators are responsible for handling the user queries, by first using the signatures of other mediators to compare data sources, then routing the query to those mediators that seem appropriate, and finally collecting and delivering the resulting data. This model slightly diverges from Zhang's in that it takes into account the aggregation work load and omits lateral connections between aggregators and databases.

Figure 1 shows an example trace of how a particular organization using this approach processes a single query. The process begins when a user query is sent to a mediator (1). The mediator then queries a number of other mediators to determine if they are appropriate to handle the user query (2). After the responses are sent (3) and collected, a subset of those searched are selected based on their reported collection signatures (4) and notified that they should handle the query (5). In this case, Mediator_R was selected, while Mediator_L was not. The user query is then propagated down all branches of the mediators' hierarchies (6, 7), until the terminal databases at the leaves are reached. Each database processes the query (8) and reports it back to its immediate manager (9). These transmission and processing activities will occur in parallel with others taking place in the organization. Intermediate aggregators will wait until all subordinates have responded, and then consolidate the results (10) before delivering the information up the next level in the hierarchy (11). Mediators perform a similar consolidation step (12). Any mediators that were selected to handle the query report their results back to the originating mediator (13), which performs a final consolidation step (14) before delivering the final response to the user (15).

This organizational design provides several advantages. The use of collection signatures to model the contents of a number of individual sources can dramatically reduce the number of agents that must be searched and queried. The use of hierarchies introduces parallelism into the query distribution process. These same hierarchies also distribute the communication and processing load.

At the same time, if the structures are poorly designed they can lead to inefficiencies. A single collection signature, which must be bounded by size to be efficiently used, can become unacceptably imprecise if the set of sources it models is large or extremely diverse. This can cause data sources to be overlooked, potentially reducing the response quality. If the data sources are distributed across many different mediators it may require a more extensive search and query process to obtain a high quality result. Whenever a hierarchy is used, there also exists a tension between the width and height of the structure. Because each agent is a bounded resource, very wide structures can lead to bottlenecks, as particular individuals with high in-degree may become overwhelmed by the number of interactions. Very tall structures can be slow or unresponsive, as the long path length from root to leaf increases latency.

An organizational model for this domain was created using ODML [4]. It is possible to use such a model as part of a larger search process to determining the most appropriate organization of agents and databases, given the desired characteristics of the system, the provided characteristics of the environment and the tradeoffs we have presented here. Like prior models created with this language, this model uses notions of roles, a task environment and performance constraints. However, other underlying phenomena that must be captured are significantly different, and drive the shape of the organization in different directions. These include statistically predicting the results of the source discovery process, determining how the information contents of a hierarchy affect its expected load and approximating the effects of increased signature uncertainty caused by summarization. This paper will focus on these latter characteristics; the reader can refer to [4] for a description of more domain-independent characteristics this model also possesses. In the discussion below, italicized terms refer to specific characteristics present in the organizational model.

3. MODELING IR CHARACTERISTICS

Like the working system it represents, there are many facets to the model. This includes how the collection information signatures are generated, a description of the query and response propagation model, a detailed model of the system's response time, and how all these features are combined into a utility value. Although each can be modeled as a particular, distinct characteristic of the system, in reality they interact in complex ways. The tensions that arise in the resulting model embody the tradeoffs and decisions that must be made when designing the organization.

3.1 Data Sources and Collection Signatures

To correctly estimate work levels in the system, we must first know the type and quantity of information that a source may provide. Different organizations may be necessary if the available information for that topic is concentrated in one spot, or if it is distributed across many separate sources. In this model, we are concerned with a single class or topic of information. This is modeled by specifying the total amount of information owned by a source, along with the fraction of which that is relevant to the topic. Aggregators and mediators, which have no information of their own, derive these values as the sum of the information that exists in the *sources* that exist below them. Ultimately, this is used to calculate the response size of the mediator, the total amount of relevant information that will be searched while processing a query.

This raw information is only half the picture, however, as the number of queries that a mediator receives is not dependent on the actual amount of data it manages, but on the data that others *perceive* it to manage. The IR search process is based on the mediator's collection signature, which is generated from the information in its hierarchy. Ideally, this would be a perfect synopsis. In practice, the signature's accuracy may be skewed by the technique used to generate it, or because of abstraction inherent in the aggregation process. This factor is taken into account in the simple recursive calculation of *perceived_response_size* for mediators ($m.prs$), aggregators ($a.prs$) and databases ($d.prs$).

$$
\begin{aligned}
m.prs &= \sum_{s \in m.sources} s.prs * aggregation\_factor && (1)\\
a.prs &= \sum_{s \in a.sources} s.prs * aggregation\_factor && (2)\\
d.prs &= response\_size && (3)
\end{aligned}
$$

3.2 Probabilistic Search and Query

The query load incurred by a mediator, and by relation any sources beneath it, is dependent on the number of queries that mediator is asked to service. This value depends on the mediator's perceived value, the average number of queries arriving in the system, the number and value of competing mediators, and how many mediators are used to answer a query. To estimate this, we must first determine the relative rank ordering m_r of the mediator in question m, and the number of mediators R_r that share that ranking.

$$
\begin{aligned}
m_r &= 1 + \Big(\sum_{k \in \mathcal{O}.mediators} 0^{max(m.prs - k.prs, 0)} \\
&\qquad - 0^{abs(m.prs - k.prs)} \Big) && (4)\\
R_r &= \sum_{k \in \mathcal{O}.mediators} 0^{abs(m_r - k_r)} && (5)
\end{aligned}
$$

Where $\mathcal{O}$ is the high-level organization and $m.prs$ is the mediator's *perceived_response_size*. The summation term will equate to 1 when the competing size is higher, and 0 when lower. Thus, the highest ranked mediator will be 1, followed by 2, and so forth. Mediators with the same value will have the same ranking. Using this information, it is possible to compute the probability $P(m)$ that a mediator m will be selected to service query Q.

$$
\begin{aligned}
P(m) &= \frac{s}{|M|} \frac{1}{\binom{|M|-1}{s-1}} \Big(\sum_{i=0}^{q-1} \sum_{j=0}^{\min(s,R_r)-1} \binom{m_r - 1}{i} \binom{R_r - 1}{j} \\
&\qquad \binom{|M| - m_r - R_r + 1}{s - i - j - 1} \min\Big(1, \frac{q-i}{j+1}\Big) \Big) && (6)
\end{aligned}
$$

Where $|M|$ is the total number of mediators, s is the number of mediators that will be searched and compared, and q is the number of mediators that will be given the query. Equation (6) models the search process and subsequent mediator selection that will take place when a query is received by the system. In this particular domain, some subset of the available mediators will be searched, and ranked based on their collection signatures. Using these ranks, a subset of those searched will actually be selected to service the query. This is a common strategy employed by agent systems, so it is worth discussing the equation in greater detail.

First, assume that all mediators may be initially searched with equal probability, and that selection from a set of equally-ranked mediators is done uniformly. The probability that mediator m is searched, which depends on the total number searched and the total number of mediators, is $\frac{s}{|M|}$. The nested summations count the total number of sets of remaining mediators that both could be searched and would result in m receiving the query. A ratio of this total to the number of possible mediator combinations from the search $\binom{|M|-1}{s-1}$ provides the final desired probability. The summations iterate over the various ways in which the mediator search set might be composed. On each loop, a value is selected for the number i of higher ranked mediators and j of equally ranked mediators that will exist in the set, the remainder being made up of lower ranked mediators. There are $\binom{R_r-1}{j}$ equal valued mediators competing for the available query slots, and the final ratio is calculates the fraction of those that might contain m. These equations are used to determine the final *query_rate* for a particular mediator, and its *query_probability* which is used elsewhere in the model.

An example organization exhibiting these features is shown in Figure 2. In this instance, there are four mediators, one with three sources, two with two sources each, and one with a single source. All databases in this model have an equal amount of topic data, so a ranking of $\{1, 2, 2, 4\}$ can be determined among the mediators respectively, as shown in the model. In addition, there are three other mediators in the organization that contain an insignificant amount of topic data and are not graphically shown. These "other" mediators are significant because they can potentially distract the search process, resulting in a decrease in expected utility. The *environment* node shows that the *search_set_size* in this instance is set to 5, indicating that the collection signatures of five other mediators will be searched. The *query_set_size*, the number of mediators from the search set that will actually be queried, is set to 3. Therefore, as the number of "other" mediators grows, the chance that one of the relevant mediators will be found and subsequently queried decreases. The value of $|M|$ in Equation (6) is the sum of the relevant mediators and these other mediators. Together these data allow the calculation of $P(m)$, the *query_probability* for mediator m. These are used to compute the organization's *response_recall*, and ultimately affect the utility of the organizational structure.

To test this formulation, a set of simulation trials were performed, and the observed response recall compared to the predicted value for each scenario. The environment consisted of six mediators and nine databases, and each trial consisted of 100 queries from a simulated user to a random mediator in the organization. The first mediator had four of the databases below it, the second had three and the third had two. The remaining three mediators with no appropriate data sources served as distractions. The *perceived_response_size* for each mediator was proportional to the number of databases it had access to. In the trials, both the number of mediators that were searched for, *search_set_size*, and the number of mediators that were queried, *query_set_size* ranged from 1 to 6. A graph comparing the values predicted by the ODML model and the empirical results are shown in Figure 3. As expected, when the search size is

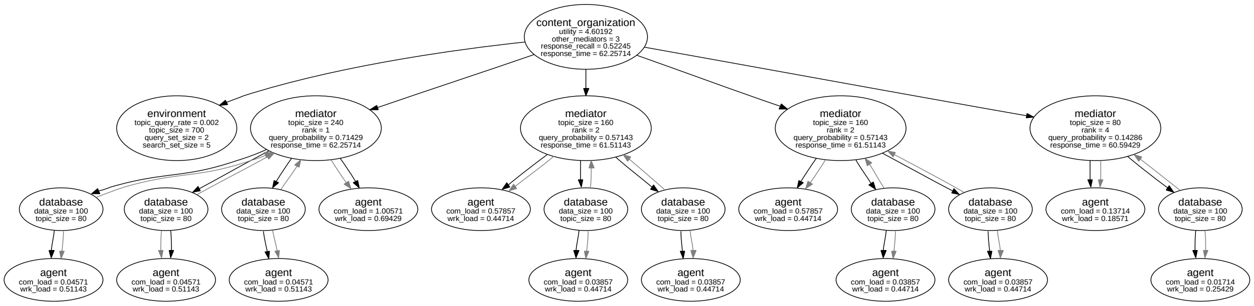


Figure 2: An information retrieval instance with variously ranked mediators.

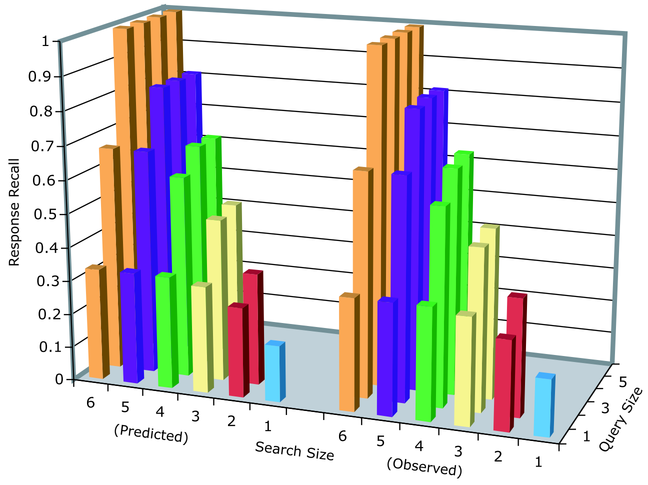


Figure 3: A comparison of the predicted and empirical response recall values as the search and query sizes are varied.

small, the recall suffers, because it is less likely a good information source will be found. The *query_set_size* has a similar but lesser effect. This shows that the predictions were quite accurate in most cases, with an average of 0.9% error over all cases. Experiments not shown here with other designs produced similar results.

The relationships described here are a good example of how changes to the organization can indirectly affect the characteristics of many, potentially distant parts of the structure. In this case, the perceived, relative quality of a mediator affects the ranking of all other mediators in the organization. These rankings affects query load, which affects the load imposed on the agents, which can affect the response time of a mediator's hierarchy as a whole. Thus it is possible for a single source added to some segment of the organization to dramatically affect agents with which it does not obviously interact. These effects can be subtle yet important, motivating the need for a model capable of representing them.

3.3 Queuing Response Time

The response time of the IR system is the amount of time that elapses between a user query and the system's response. This particular characteristic is clearly important from an evaluation standpoint, as it captures an easily observable phenomena that is important to the end user. Like the probabilistic query model, the response time is intimately tied to the structure of the organization. Several characteristics affect this value. For example, each communication event incurs some message transit latency. The query processing by the databases, and the aggregation performed by both the aggregator and mediator will take some variable amount of time. The latter two entities must also wait for slowest of their information sources before they can themselves respond. Finally, because multiple queries can exist simultaneously in the network, additional delays at individual agents can be incurred when a query must wait for existing processing to complete. The model of this system draws upon techniques from probability and queuing theory. This section will first discuss why simpler techniques are insufficient and derive the actual solution in stages, gradually incorporating new elements as deficiencies are recognized.

The mediator's *query_rate* characteristic predicts the arrival rate of queries to that mediator based on the probabilistic model given above. One can infer that responses will, on average, be returned back to the user at this same rate. The response rate cannot be faster, because the system would eventually run out of queries to process. If the response rate were slower, the number of queries in the system (along with the expected response time) would tend to infinity as new queries encounter an ever lengthening queue of existing queries upon arrival. Of course, this is not an impossible situation, just undesirable, so we must include constraints in the model that specify that the possible rate at which tasks are serviced must be greater than or equal to the rate at which they arrive. Given these constraints, we can assume that the query rate will equal the response rate. We can furthermore assume this is the case for all agents in the hierarchy by analogous reasoning, after observing that the arrival and response rates of one agent dictate the complementary rates of the other agents they are attached to.

More concretely, the existing model assumes that queries arrive to the mediator with a Poisson arrival distribution and mean rate *query_rate*. The model also specifies that tasks arrive with rate *arrival_rate*, where each task is a query and *arrival_rate* = *query_rate* in this instance. This means that the amount of time between subsequent tasks will be a random value sampled from an exponential distribution with parameter *arrival_rate* — on average, a new query will arrive at the mediator every $arrival\_rate^{-1}$ milliseconds.

After some amount of time elapses, during which the query makes its way down through any aggregators, the databases themselves will receive the query. By the logic given above, we can assume that they arrive at rate *arrival_rate*. Each database is also associated with a *service_rate*, which reflects its ability to complete queries given to it. When this new query arrives there may be previously received queries currently being processed or waiting to be processed by the database. Because we assume first-in, first-out processing, the amount of time the new query must wait before being addressed will depend in part on these existing queries.

We can exploit existing techniques from the field of queuing theory [6, 8] to analyze how long the wait will be, an approach that has recently proved successful in other MAS models [3, 9]. For example, the database can be described as a *M/M/1* queue. This model assumes a Poisson task arrival rate (i.e., *arrival_rate*) and service rate (i.e., *service_rate*), and a single processor (i.e., the agent performing the database role and the resources under its control). From this information, one may immediately determine the expected *service_time* of the database, using the formula [8]:

$$service\_time = \frac{1}{service\_rate - arrival\_rate}$$

Unfortunately, this single expected value is not sufficient, for reasons that will become clear below. Instead, the model uses this same basic information to compute approximations of the probability density function (pdf) and cumulative distribution function (cdf) of the service time characteristic. These functions represent richer forms of the same waiting time knowledge, because they preserve the statistical character of the phenomena, rather than just a sample mean. The pdf $f_M(x, \lambda)$ and cdf $F_M(x, \lambda)$ of the *M/M/1* queue's waiting time distribution are [2]:

$$\begin{aligned} f_M(x, \lambda) &= \lambda e^{-\lambda x} \\ F_M(x, \lambda) &= 1 - e^{-\lambda x}, \end{aligned}$$

where $x \geq 0$ and $\lambda = service\_rate - arrival\_rate$. The model maintains this information as a discrete list of sampled points, which are calculated dynamically from the two underlying functions. In particular, it defines the following two lists:

$$\begin{aligned} pdf\_list &= [f_M(x_0\, d\_step, \lambda), \ldots, f_M(x_n\, d\_step, \lambda)] \quad (7) \\ cdf\_list &= [F_M(x_0\, d\_step, \lambda), \ldots, F_M(x_n\, d\_step, \lambda)], (8) \end{aligned}$$

where $x_n = n$ and $(0 \leq n < \frac{d\_range}{d\_step})$. *d_range* represents the upper bound on the sampled points, while *d_step* is the stride length between points. Note that, if we wish, we can compute an expected *service_time* for the database from this data, using the conventional definition of expected value:

$$service\_time = \sum_{x=1}^{d\_range/d\_step} (x\; pdf\_list[x]\; d\_step^2) \qquad (9)$$

Ignoring the aggregator for the moment, we will focus our attention on the mediator. Recall that it must wait for responses from all its information sources before progressing, which directly ties its *service_time* to those of the information sources below it. One approach would find the average such *service_time*, and use that to estimate that component of the mediator's service time. However, this metric is actually a biased approximation of the service time. Recall that databases are modeled as Poisson processes. Although they do have a mean service time, most of the time a greater or lesser value will be observed in practice. If there is just one database to wait for, the sample mean would be a sufficient. However, if there are two or more databases, multiple samples will be made, effectively increasing the chance that a greater-than-mean value will be observed. This skews the response time distribution so that the aggregate waiting time is no longer Poisson. This effect can be seen in Figure 4, which shows how the waiting time distribution of a processing agent changes with the number of databases below it. Notice that as the number of databases increases the distribution shifts to the right, which is consistent with our intuition.

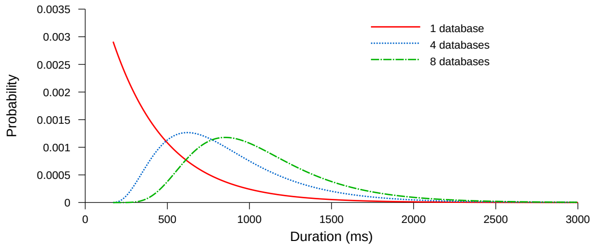


Figure 4: A comparison of the waiting time distributions for differently sized sets of databases.

The underlying issue is that the mediator in this system must wait for the *slowest* responder, which means that its service time will be dependent on the *maximum* service time of those below it, not simply the average. This is the motivation for explicitly representing the pdf and cdf of the database response above. Given this more detailed information, it is possible to determine what the mediator's actual waiting time distribution is. A branch of probability theory known as *order statistics* is useful here. Assume that we have n samples from some distribution X, $(X_1, \ldots, X_n)$. $X_{(k)}$, the kth smallest sample from this set, is known as the kth order statistic [1, 7]. The nth or *maximum order statistic* of the distribution is the expected maximum value in the set of samples. This corresponds to the amount of time the mediator is expected to wait before all responses have been received.

The example statistic given above is known as the independent, identically distributed (iid) case, because all n samples are from the same random variable X. Unfortunately, this is not the case seen in the IR system. First, we do not assume that all databases have the same service rate. Second, different aggregation structures with different heights and widths will also produce different waiting time distributions. The model does make a simplifying assumption, that the various samples are independent. Together, this is known as the independent, non-identically distributed (inid) case. The model generates the pdf $f_{(n)}$ and cdf $F_{(n)}$ sample distributions of the nth order statistic for the source service time using the following functions from [1] and [7]:

$$\begin{aligned} f_{(n)}(x) &= \left[\prod_{i=1}^{n} F_i(x)\right] \sum_{i=1}^{n} \left(\frac{f_i(x)}{F_i(x)}\right) \qquad (10) \\ F_{(n)}(x) &= \prod_{i=1}^{n} F_i(x), \qquad (11) \end{aligned}$$

where f_i and F_i represent the pdf and cdf of the ith sample, respectively (i.e., the service time distribution of the ith information source). Sample lists are generated for these two distributions in the same manner shown in equations 7 and 8.

The mediator itself is not simply a pass-through, but must process and aggregate the resulting data as well. Just as with the processing of queries by the database, the processing of the results by the mediator also takes time, potentially causing newly arrived results to wait until the mediator can devote attention to them. Thus, the mediator can also be viewed and modeled as a queue. In this case we will assume it is also a *M/M/1* queue, with an arrival rate consistent with the argument presented earlier. The service rate of the mediator depends on the number of responses it receives, which depends on the number of information sources below it. The mediator's pdf and cdf can also be produced using Equations 7 and 8, with $arrival\_rate = query\_rate$, $service\_rate = response\_service\_rate/num\_sources$, and a Poisson rate of $\lambda = arrival\_rate - service\_rate$.

To complete the model we must determine the behavior of the total service time that combines these two activities. This can be accomplished by recognizing that this service time will be the sum of the times exhibited by these two random variables, since the local processing phase takes place after all results have been received. The total service time pdf f_C and cdf F_C can then be determined by finding the convolution of the corresponding distribution functions, which has the general form:

$$f_C(x) \quad = \quad \sum_{i=0}^{d\_range/d\_step} f_s(i) f_l(x-i) d\_step \qquad (12)$$

$$F_C(x) \quad = \quad \sum_{i=0}^{d\_range/d\_step} f_s(i) F_l(x-i) d\_step \qquad (13)$$

For the mediator, f_s would be the aggregate information source pdf given in Equation 10, while f_l and F_l would be the pdf and cdf of the waiting time for the local *M/M/1* queuing process.

Both these convolution equations and those used earlier to compute the maximum order statistics make no assumptions about the underlying distributions they reference. Because of this, any other queuing model can be substituted for the *M/M/1* queues used in these roles, so long as it can be characterized or approximated through a closed form formula using the mathematical primitives supported by ODML. One could also directly provide a complete discrete distribution in its place.

The model can now compute the expected *service_time* of the mediator by using the same expression previously shown in Equation 9, coupled with the cumulative overhead incurred by the message transit times of the query and result propagation process. This can be used along with the mediator's *query_probability* to predict the *response_time* distribution of the organization as a whole.

Note that Equations 10-13 are recursive, in that they rely upon both the pdf and cdf distributions of the sources below the mediator. The equations make no assumptions about the form of those distributions, so they can be used both when the information source is a single database or an arbitrarily complex aggregator hierarchy. This same assumption also allows Equations 10 and 11 to be used to compute the pdf and cdf distributions for the aggregator itself. The recursive definition will terminate in the exponential response distribution exhibited by the databases. The aggregator also performs response aggregation, which can be approximated with a suitable queuing model. The expressions given in Equations 12 and 13 are again used to combine these two characteristics to determine the aggregator's total service time pdf and cdf distributions.

The final aspect that must be taken into consideration is the effect that multiple roles have on performance. This will be approximated by weighting the *service_rate* for each role based on the proportion of local processing time it is expected to receive. In particular, if λ_i and μ_i are the original *arrival_rate* and *service_rate* for an agent's ith role, let the *effective_service_rate* (μ_{i_E}) of that role be:

$$\mu_{i_E} = \mu_i \frac{\lambda_i/\mu_i}{\sum_{r=0}^{\#roles} \lambda_r/\mu_r} = \frac{\lambda_i}{agent.work\_load} \qquad (14)$$

where *agent.work_load* is the location in the model where the aggregate demand is stored. This expression uses each role's arrival and service rates to first determine the expected proportion of time the agent will be busy, and then to scale the individual service rates accordingly. The role's arrival rate remains unchanged.

The results of a several sample runs are shown in Figure 5. Each scenario measures the response time performance of a different IR organizational design, by submitting 1000 queries to it in a Poisson fashion as described previously. The organizational design of each scenario is depicted on the left, along with the predicted and empirical response time data on the right. The solid line represents the distribution of response times predicted by the ODML model, while the dashed line indicates the observed frequency of individual response times in the simulation. A bin width of $W = 2(IQR)N^{-1/3}$ was used to group the empirical response times, where N is the number of trials and IQR is the interquartile range of the data (the difference between the 75th percentile and 25th percentile of the data). This is the Freeman-Diaconis rule, as discussed in Izenman's bin width strategy analysis [5].

As can be seen in the performance graphs, the ODML model does a good job of predicting the response time distribution of the different organizational designs. Additional response time trials were performed for organizations with three agents [1M,2D], five agents [1M,4D], 10 agents [1M,2A,7D], and 14 agents [1M,3A,10D], with similar results. The coefficient of determination R^2 $(= 1 - \frac{(y-\hat{y})^2}{(y-\bar{y})^2})$ was calculated for each scenario, which estimates how much of the observed behavior can be explained by the model [2]. R^2 was greater than 0.8 for all tested scenarios, where a value of 0.7 or above is considered good for this statistic.

The effort taken to preserve the underlying probability distributions in this aspect of the model has other benefits, besides being necessary to accurately model the response time behavior. This same information can be used to support high-level behavioral constraints. For example, a constraint can be defined using the mediator's cdf that places an upper bound on the probability that a particular response time is exceeded. The pdf can also be used to determine the average response time as shown above, which will be used to define the organizational utility in the following section. By explicitly capturing the "fuzzy" nature of the running system's performance, these richer statistics allow the designer greater control over the

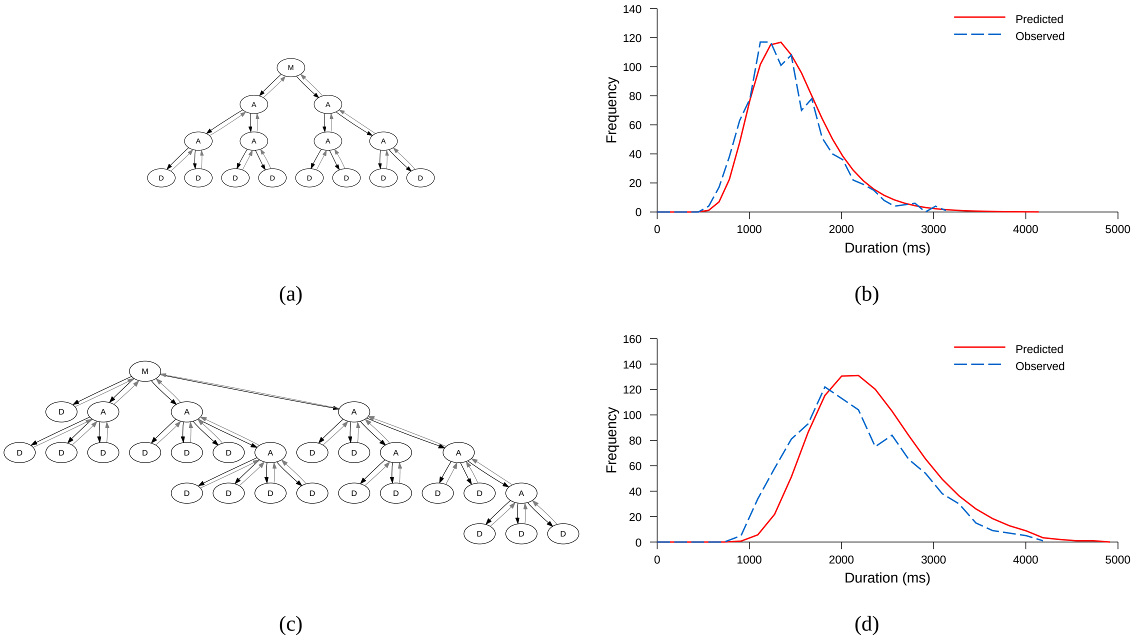


Figure 5: A comparison of the predicted and observed response time distributions in organizations with (a,b) fifteen [1M,6A,8D], and (c,d) twenty-eight [1M,7A,20D] agents. In (a,c), node M is a mediator, A are aggregators, and D are databases.

evaluation and output of an organizational design process.

3.4 Determining Design Utility

Both *response_recall* and *response_time* are used by [10] to evaluate the performance of the system. These two metrics are combined in the ODML model to produce a single *utility* field, which can then be used to compare and rank candidate designs during a search process. In this case, recall is more important than response time, so a multiplicative factor is applied to the recall value, after which the response time is subtracted out:

$$utility = response\_recall * 1000 - response\_time/10$$

Recall is the proportion of the possible information that was reported ($[0 \ldots 1]$), while time is measured in milliseconds ($[0 \ldots \infty)$, generally in the thousands). The normalization terms cause this formulation to generally favor quality over speed, and instances with equal recall will be differentiated by their response time. An arbitrary utility function could be substituted as needed.

Figure 6 shows how utility is affected by the expected user query rate. Optimal values are shown in bold. This figure shows all eighteen possible organizations that are possible in a six database environment with a maximum height of three and a minimum of two subordinates per node. The *search_set_size* and *query_set_size* of each organization is set to one. Organizations have zero utility at a given query rate when the query arrival rate exceeds the organization's service rate, resulting in an infinite length queue.

The single-level, single-mediator organization number 1 is predicted to be optimal when the query rate is 0.5 or less (i.e., less than one query every other second). This is intuitive, because the slow query rate avoids queuing delays, causing the response time to be dominated by the height of the organization.

As the query rate increases, first organization 8, then number 9 and finally number 11 become optimal, as the highly-connected mediator in organization 1 becomes an increasing bottleneck. In contrast to most of the competing designs, organizations 8, 9 and 11 are all balanced (as is number 1). In an unbalanced organization, the segment with greater load tends to dominate the response time because the final result must wait for the slowest responder. These three designs avoid this by evenly spreading the load among participants.

Organization 9 outperforms 8 under higher loads because the database aggregation is better distributed, which reduces the chance that any one node will observe high queuing delays. By having three aggregation points of size two rather than two of size three, the range of likely durations is reduced (see Figure 4).

Because the search and query sizes are set to one, the multi-mediator organizations 11-18 exhibit different recall characteristics than 1-10. At most one mediator is searched in these tests, so at most one will contribute to the user's query. The recall of organizations with two mediators is roughly half that of those with one, while organization 18 with three mediators has but a third of that recall. This correspondingly degrades the organization's utility.

The benefit the multi-mediator designs offer is increased robustness to high work loads. Where no single-mediator organization can handle more than six queries per second, all eight multi-mediator designs can obtain utility with at least seven queries per second. This is because the smaller search size reduces the query rate any individual mediator sees. The aggregate demand on the system is lower, which reduces the growth rate of individual agents' queues, which allows the system as a whole to tolerate higher query rates (albeit with lower recall).

These tests show the spectrum of tradeoffs that can be made in

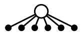
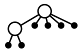
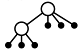
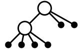
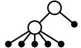
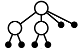
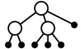
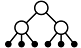
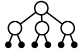
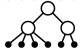
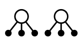
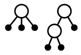
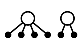
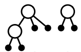
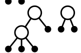
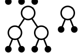
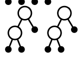
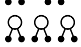
Organization	Query Rate									
	0.1	0.25	0.5	1	2	3	4	5	6	7
1. 	**725**	**723**	**719**	**710**	672	544	0	0	0	0
2. 	715	714	711	704	681	621	0	0	0	0
3. 	716	715	712	707	691	664	598	0	0	0
4. 	715	714	712	706	691	664	598	0	0	0
5. 	713	712	709	702	679	621	0	0	0	0
6. 	710	709	706	701	687	662	599	0	0	0
7. 	711	710	708	704	693	677	651	597	465	0
8. 	712	711	709	705	**695**	**680**	**657**	612	482	0
9. 	705	704	702	699	690	677	**657**	**619**	**517**	0
10. 	710	709	707	702	689	665	604	0	0	0
11. 	379	379	379	378	378	378	377	376	375	**374**
12. 	374	374	374	374	373	373	372	372	371	370
13. 	379	379	378	378	378	377	376	375	373	371
14. 	374	374	374	374	373	373	372	371	371	370
15. 	374	374	374	373	373	373	372	371	371	370
16. 	373	373	373	373	373	372	372	371	371	370
17. 	373	373	373	373	373	372	372	371	371	370
18. 	252	252	252	252	251	251	251	251	251	251

Figure 6: The utility predicted for the range of possible six-database organizations when the query rate (queries per second) is varied. Mediators and aggregators are shown as hollow circles, while the solid databases form the leaves. Higher is better, optimal values for each rate are shown in bold.

this design in different environments. Organization 1 is fast but quite centralized. To avoid the increasing chance of bottlenecks, organization 9 trades off response time and 11 trades off response quality. Despite these different tactics, the underlying pressures are the same — note the parallels between organizations 11 and 1, which are optimal at opposite ends of the spectrum.

Additional experiments not shown here have also explored the consequences of setting the *search_set_size* and *query_set_size* so all mediators will be employed for each query. The resulting designs essentially undo the tradoffs made by the multi-mediator organizations 11-18. The response recall in each is brought up to the levels of the single mediator designs, but the response times increase correspondingly. For example, the utility of organization 11 is 721 at rate 0.1, 695 at rate 2, and 0 at rates higher than 5. The end result is that these designs rarely offer benefits not found in their single-mediator counterparts.

4. CONCLUSIONS

This paper has discussed some of the interesting, organizationally-driven characteristics that exist in a working information retrieval system. A model of that system created in the ODML framework was described that captures these behaviors, including a utility-driven search process in the agent network and the consequences of queuing on response time in a distributed work flow. This demonstrates how existing mathematical techniques can be successfully incorporated into organizational models, and how the detailed predictions they make can be used to predict the utility of design alternatives.

In particular, the search behavior of this system as well as the hierarchical control it employs are not uncommon in the field of multi-agent systems. This suggests that techniques we have used from the fields of queuing and probability theory can be used elsewhere with similar results.

More importantly, this work demonstrates how significant domain-specific characteristics can be affected by choices made in an organizational design. If such characteristics are not accounted for in the design process, or if they are approximated to the degree that important interactions are lost, then the ramifications of decisions that affect those characteristics may also be lost. We believe this will affect the quality of any resulting design, as those interactions are real and must be considered. This fact motivates the detailed view of organizational behavior that is expressed in this paper.

5. REFERENCES

[1] H. A. David. *Order Statistics, 2nd Ed.* Wiley, 1981.

[2] J. L. Devore. *Probability and Statistics for Engineering and the Sciences (Fourth Edition)*. Wadsworth, Inc., Belmont, CA, 1995.

[3] N. Gnanasambandam, S. Lee, N. Gautam, S. R. T. Kumara, W. Peng, V. Manikonda, M. Brinn, and M. Greaves. Reliable MAS performance prediction using queueing models. In *Proceedings of the IEEE Multi-agent Security and Survivability Symposium (MASS)*, 2004.

[4] B. Horling and V. Lesser. Analyzing, Modeling and Predicting Organizational Effects in a Distributed Sensor Network. *Journal of the Brazilian Computer Society, Special Issue on Agents Organizations*, pages 9–30, July 2005.

[5] A. J. Izenman. Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, 86(413):205–224, March 1991.

[6] L. Kleinrock. *Queueing Systems. Volume I: Theory*. John Wiley & Sons, New York, 1975.

[7] Reiss, R.D. *Approximate Distributions of Order Statistics*. Springer-Verlag, New York, NY, 1989.

[8] S. Ross. *Introduction to Probability Models*. Academic Press, Boston, MA, 5th edition, 1993.

[9] M. Sims, J. Kurose, and V. Lesser. Streaming versus Batch Processing of Sensor Data in a Hazardous Weather Detection System. In *Proceedings of Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2005)*, September 2005.

[10] H. Zhang and V. Lesser. A Dynamically Formed Hierarchical Agent Organization for a Distributed Content Sharing System . In *Proceedings of the International Conference on Intelligent Agent Technology (IAT 2004)*, pages 169–175, Beijing, September 2004. IEEE Computer Society.