Strategic Web-Service Agreements

Sudarshan S. Chawathe

Department of Computer Science
University of Maine
Orono, ME 04469, USA
`chaw@cs.umaine.edu`

Abstract— **This paper addresses issues of strategy in Web-service composition, and inter-site collaboration in general. The results are useful for both human and artificial agents of a site who are responsible for determining Web-service agreements that are most profitable for that site. We discuss three specific questions in this area. (1) How should the profit resulting from a composition of Web-services be divided among the participants? (2) How can we counter the risk of service-providers misrepresenting their services in an attempt to gain a larger share of the profit? (3) Are stable configurations guaranteed or feasible when each service-provider's decisions on how to collaborate (permit compositions) are guided solely by the goal of maximizing its profit?**

I. INTRODUCTION

We study the strategic aspects of agreements among Web-service providers. Consider a collection of autonomous sites that offer Web services of various sorts. Throughout this paper, we use the term *site* to refer to an autonomous entity that offers one or more Web services. The simplest realization of this concept is a Web-service daemon running on a single machine connected to the Internet. However, we also use the term to refer to a distributed infrastructure that is logically one autonomous entity offering a service. As an example of a site of the former kind, consider one of the many *XTide* [17] tide-prediction servers on the Web. An example of the latter kind is the collection of Google APIs [19], which may be implemented using a substantial distributed infrastructure.

One of the main advantages of the Web-service model is that services can be composed in a flexible yet systematic manner in order to address needs that may not have been anticipated by any of the individual service providers. Given the importance of composition, there has been a substantial amount of recent work on the topic, addressing questions such as how Web services are specified in a standard and expressive manner, how they are discovered, and how they are orchestrated. Several methods for composing Web services automatically or semi-automatically have been proposed. The focus of such work is on the standards, query-language, or logical and ontological aspects of composition. While these are important aspects that need continuing attention, it is also important to consider the strategic aspect.

The strategic aspect of composition concerns several questions arising from the reasonable assumption that many Web services are likely to be operated as business units and therefore will enter into service-composition agreements in a manner designed to maximize their own profit. For example, a site that provides weather reports may find it profitable to form a partnership with a site that provides local news in order to gain access to additional customers. However, the news site may determine such a partnership is not in its best interest because of the risk of losing advertising revenue from a competing weather site.

We note an important difference between our model in this paper and that used by some prior work: We assume that compositions or other interactions between sites are controlled by the participating sites themselves. The mechanics of this control are not important, and may include technological or legal components. For example, a site may forbid compositions with another by the use of a computer-readable specification, by use of network filtering tools, or by means of a user agreement that forbids certain actions. While some services may permit unrestricted composition and other uses, it is likely that high-value services will include restrictions on compositions This strategy is commonly employed by many current sites. For example, there are explicit restrictions on how the package-tracking features provided by FedEx may be used [16]: "Use of fedex.com to provide information to or prepare shipments by or for the benefit of third party shippers is expressly prohibited." UPS places similar restrictions on use of its Web site [37]. The Amazon Web Services API is also subject to significant restrictions [7].

As an example of a problem arising from the actions of sites that wish to maximize their expected profit in a Web-service scenario, consider the screenshot in Figure 2 on page 4. In terms of service composition, this example is not very exciting, as it consists of a very simple service interface (keyword and other search, listing of remote items with prices, etc.). However, it is evident that some of the participating sites have reported artificially low prices in an attempt to increase the number of Froogle users visiting their sites. This example is discussed further in Section III, which addresses the problem of how to cope, at a system level, with the problem of sites potentially misrepresenting their services.

The methods presented in this paper are useful to a human

This work was supported in part by the U.S. National Science Foundation with grants IIS-9984296, IIS-0081860, and CNS-0426683.

principal charged with the task of pursuing or preventing strategic Web-service composition agreements between sites. However, they are even more useful when Web-service policies are dynamically determined by an artificial agent, perhaps changing in response to changing competition, perceived demand, or other technical and business factors. For example, an implementation of the method for distributing shared profits described in Section II permits the determination of payoffs from dynamically generated service compositions.

The next three sections address three questions related to strategic composition of Web services: Section II: How should the profit resulting from a composition of Web-services be divided among the participants? Section III: How can we counter the risk of Web service-providers misrepresenting their services in an attempt to gain a larger share of the profit? Section IV: Are stable configurations guaranteed or feasible when each service-provider's decisions on how to collaborate (permit compositions) are guided solely by the goal of maximizing its profit? We discuss related work in Section V and conclude in Section VI. In order to keep the presentation manageable, our discussion often centers on a small and simple example introduced in the next section. However, our methods are equally applicable to larger and more complex scenarios.

II. PROFIT-DISTRIBUTION

Suppose a collection of sites that interact with each other using Web services generates some profit as a result of these interactions. How should this profit be distributed among the participants? Perhaps the simplest solution is to allot an equal share of the profit to each participant. However, this solution is not compelling as it completely ignores the differing contributions of the participants. Intuitively, it seems fair that a site should receive a share of the profit that is proportional to its *contribution*. The key question is how we quantify the intuitive notion of a site's contribution. In some cases, the answer may be readily evident. For example, if a purchase order of 10 copies of a book is fulfilled by a collaboration among three bookstores that supply 5, 3, and 2 copies, all at the same cost, then it seems reasonable to divide the profit in the proportion $5:3:2$. However, most situations involving a collaborating set of Web-service sites are far more complex, and do not permit the contribution of a site to be extracted with such ease. Intuitively, complications arise because the contribution of a site is not independent of the participation of other sites. For example, the contribution of a site that provides rental-car information for an airport is likely to be higher when the contribution of another site results in a booking of a flight to that airport.

We illustrate the above idea using an extended example that is used, with some modifications, throughout the rest of this paper. Consider the situation suggested by Figure 1, consisting of three *merchant* sites, m_1, m_2, m_3, that sell some products and two *store* sites, s_1, s_2 that provide search and listing facilities (Web store-fronts, portals, search engines, etc.). The products sold by each merchant are listed in Table I.

Merchant	Products Sold
m_1	A
m_2	B, C
m_3	C, D, E

TABLE I

PRODUCTS SOLD BY THE MERCHANTS IN THE EXAMPLE OF FIGURE 1.

The indexing and listing services provided by the store sites s_1 and s_2 are not uniform over the merchants and their products. Therefore, the probability that a customer using these sites finds products at the merchant sites varies based on the store, merchant, and product. This situation is summarized in Table II.

Store	Product	Merchant	Success
s_1	A	m_1	0.9
s_1	B, C, D, E	m_1	0.0
s_1	B, C	m_2	0.8
s_1	A, D, E	m_2	0.0
s_1	A, B, C, D, E	m_3	0.0
s_2	A, B, C, D, E	m_1	0.0
s_2	B, C	m_2	0.4
s_2	A, D, E	m_2	0.0
s_1	A, B	m_3	0.0
s_1	C, D, E	m_3	0.2

TABLE II

SUCCESSFUL-SEARCH PROBABILITIES FOR THE EXAMPLE OF FIGURE 1, AS A FUNCTION OF STORE, PRODUCT, AND MERCHANT. ROWS WITH MULTIPLE ENTRIES IN THE PRODUCTS COLUMN INDICATE THE IDENTICAL SUCCESS PROBABILITIES FOR EACH OF THOSE PRODUCTS.

For simplicity, we assume that the merchants have no way to sell their products other than through one of the stores and, similarly, that the stores have no products to sell other than those provided by the merchants. Suppose both store sites expect 100 customers daily, with each customer equally likely to buy one each of products A through E and, further, that selling each product yields a net benefit (profit) of \$10. The expected total daily profit that accrues from a cooperation among two or more sites in our example is summarized in Table III. The factor of 1000 reflects the benefit of 100 customers purchasing one product each, generating a profit of $1000 \times$ \$10. For the cooperating set $\{s_2, m_2, m_3\}$, the subtractive term is necessary because a customer using store s_2 may purchase the product C from either merchant m_2 or merchant m_3, but not from both. (Recall that in our example, each customer desires one of each kind of product.) A similar reasoning lies behind the other subtractive terms in the table.

The table indicates several situations in which the profit resulting from sites' contributions depends on other participants. For example, the profit from $\{m_1, m_2\}$ is 0, as is the profit from $\{s_1, s_2\}$. However, all together the four sites result in a profit of 3300. The effect of other sites' participation may be negative as well. For example, the profit for $\{s_1, m_2\}$ is 1600 and that for $\{s_2, m_3\}$ is 600; however, the collection of all four sites together produces only 2920

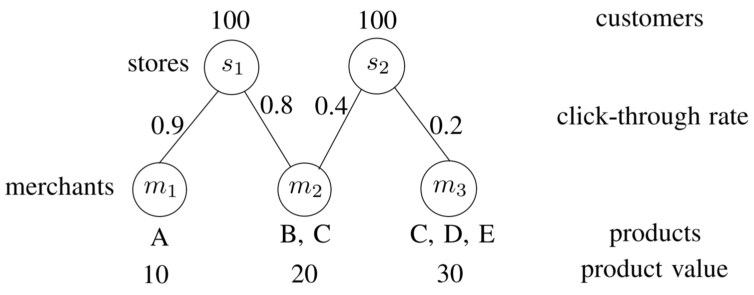

Fig. 1. A small network of stores and merchants.

Cooperating sites	Expected profit	
	calculation 1000×	result
$\emptyset$ and all singletons		0
m_i, m_j, for $i, j \in \{1, 2, 3\}$		0
m_1, m_2, m_3		0
s_1, s_2		0
s_1, m_1	0.9×1	900
s_1, m_2	0.8×2	1600
s_1, m_3	0.0×3	0
s_2, m_1	0.0×1	0
s_2, m_2	0.4×2	800
s_2, m_3	0.2×3	600
s_1, s_2, m_1	0.9×1	900
s_1, s_2, m_2	$(0.8 + 0.4) \times 2$	2400
s_1, s_2, m_3	0.2×3	600
s_1, m_1, m_2	$0.9 \times 1 + 0.8 \times 2$	2500
s_1, m_1, m_3	$0.9 \times 1 + 0.0 \times 3$	900
s_1, m_2, m_3	$0.8 \times 2 + 0.0 \times 3$	1600
s_2, m_1, m_2	$0.0 \times 1 + 0.4 \times 2$	800
s_2, m_1, m_3	$0.0 \times 1 + 0.2 \times 3$	600
s_2, m_2, m_3	$0.4 \times 2 + 0.2 \times 3 - 0.4 \times 0.2 \times 1$	1320
s_1, m_1, m_2, m_3	$0.9 \times 1 + 0.8 \times 2 + 0.0 \times 3$	2500
s_2, m_1, m_2, m_3	$0.4 \times 2 + 0.2 \times 3 - 0.4 \times 0.2 \times 1$	1320
s_1, s_2, m_1, m_2	$0.9 \times 1 + 0.8 \times 2 + 0.4 \times 2$	3300
s_1, s_2, m_1, m_3	$0.9 \times 1 + 0.2 \times 3$	1500
s_1, s_2, m_2, m_3	$0.8 \times 2 + 1.320$	2920
s_1, s_2, m_1, m_2, m_3	$0.9 \times 1 + 2.920$	3820

TABLE III

EXPECTED DAILY PROFIT $p(C)$ FOR ALL SETS C OF COOPERATING SITES IN THE EXAMPLE OF FIGURE 1 AND TABLES I AND II.

and not $1600 + 600$. In our toy example, these facts are a direct consequence of the very simple network model used. However, in general there may be no such model and the values in a table analogous to Table III may not be the result of simple underlying computations. We need a method that works in all such situations, not only the systematic one outlined in our small example.

A standard solution is the one characterized by the Shapley value [34]. An intuitive characterization of this method is that it distributes profit in proportion to the weighted average of the contribution of a site, where the average is computed over all possible configurations of other participating sites and the weights reflect the likelihoods of these configurations. More precisely, the fraction $q(i)$ of the profit that is allocated to site i in a collection of n sites is as follows:

$$q(i) = \sum_{S \subseteq C \setminus \{i\}} w(S) \cdot \Delta p(S, i), \quad \text{where} \qquad (1)$$

$$\begin{aligned} w(S) &= \frac{|S|!\ (n - |S| - 1)!}{n!} \quad \text{and} \\ \Delta p(S, i) &= p(S \cup \{i\}) - p(S) \end{aligned}$$

Consider a hypothetical process that adds sites to the set of collaborating sites in a sequential manner. The set S that indexes the summation may be interpreted as the set of sites already in the collaboration when site i joins. The additional profit resulting from site i joining at this stage of the process is quantified by $\Delta p(S, i)$, while $w(S)$ is the probability of i joining when exactly the sites in S are already collaborating, assuming all sequences of sites joining are equally likely. For our example of Figure 1, these computations are easily performed using Table III.

III. MISREPRESENTATIONS

In Section II we assumed that each Web-service site provides an accurate description of its services. For instance, we assumed that the numbers used in the example summarized by Tables I and II were accurately reported by the stores s_1, s_2 and the merchants m_1, m_2, m_3. However, this assumption is often invalid. In our example, a store may find it possible to accrue additional profit by exaggerating the number of customers it attracts. Similarly, it may be possible for a merchant to benefit by exaggerating its inventory of highly profitable products. The screenshot in Figure 2 depicts a simple but real example of this phenomenon. A Froogle [20] search for "`kitchen sink 6"`" yielded a number of products for the bargain price of \$9.99. However, a closer look reveals that the prices are not real, as suggested by the disclaimer "Display Price Inaccurate." (The links in this example lead to Amazon-affiliate merchants. The details of the mechanism used to achieve the low-price listing are unclear.)

In general, there are very many ways in which a Web-service site may misrepresent its capabilities in an attempt to improve its payoff. The example of Figure 2 is among the simplest, consisting only of misrepresenting the price. This misrepresentation is also easy to detect, at least in principle, perhaps by using a specialized Web crawler that periodically checks the prices reported by merchants with those on the merchant's Web site. In our example of Tables I and II, if store s_2 exaggerates the percentage of store visitors who make purchases, it may be possible to detect the exaggeration by monitoring traffic. However, if the reported numbers are expected long-term averages, a few observations that do

kitchen sink 6"

E.g. "digital camera" or "car near San Francisco"

Results for **kitchen sink 6"**

American Standard 7163.001.021 Silhouette collection ...
Display Price Inaccurate. Check Price at Website. Americast brand engineered material., Double bowl design., Self-rimming or undercounter ...
$9.99 XFactor

American Standard 7163.001.021 Silhouette collection ...
Display Price Inaccurate. Check Price at Website. Americast brand engineered material., Double bowl design., Self-rimming or undercounter ...
$9.99 foospecial

American Standard 7163.202.227 Silhouette collection ...
Display Price Inaccurate. Check Price at Website. Americast brand engineered material., Double bowl design., Self-rimming or undercounter ...
$9.99 foospecial

Fig. 2. A simple example of Web services providing misleading information to Froogle [20]. The search result for 6" kitchen sinks includes a large number of links to products with very low displayed prices, with the caveat "Display price inaccurate" noted in the text.

not match the store's claim may not be sufficient evidence because the store could claim, perhaps falsely, that it is simply experiencing a temporary dip in sales. In scenarios that are larger and more sophisticated than our simple examples, detection may be very difficult, or impossible. For example, consider a cooperating set of sites that permit inter-site transactions, such as those desirable for booking different components of a vacation (air fare, rail fare, hotel reservations, entertainment, etc.). The only readily observable effect of some misrepresentations (such as overstated availability of tickets) may be a high number of transactions that are rolled back. Determining the cause of a rollback and the party at fault is complicated because of the dependencies between various tasks. It is difficult to determine whether a customer abandoned a transaction because of unfavorable air fares, inadequate hotel rooms, unexciting entertainment, some combination of these factors, or simply because the customer was not seriously contemplating a purchase in the first place.

In the rest of this section, we focus on the problem of preventing misrepresentations when there is no reliable method for identifying the sites, if any, that have provided inaccurate information. In this environment, the only observable effect of misrepresentation by one or more sites is the lowering of the effectiveness of the system. In the vacation travel example, the observable effect may be a booking frequency that is much lower than what is expected based on the reported numbers.

Consider n autonomous Web-service sites $s_1, s_2, \ldots, s_n$. Each site provides one or more of the services $j_1, j_2, \ldots, j_m$. We fix our discussion on the requirements of a single class of tasks. In the simplest case, a task may specify the exact number of times each service is required. In our vacation travel example, for instance, a service that reserves rooms at hotels may be required twice, once in each of two cities on the itinerary. In general, there may be some flexibility in the number of times a service may be profitably performed. We model this situation using vectors $l = (l_i)_{i=1}^m$ and $u = (u_i)_{i=1}^m$ where l_i and u_i denote lower and upper bounds on the number of times service j_i may be profitably used. The sites provide services with varying degrees of effectiveness, as quantified by the *service matrix* defined below. The precise semantics of effectiveness are immaterial for our approach; effectiveness may represent a combination of factors such as response time, throughput, quality of data, and expected profit.

The effectiveness of the n sites for m services is represented using an $n \times m$ *service matrix* $E = (e_{ij})$, with e_{ij} denoting the effectiveness of site i on service j. The *service vector* $v_i = (e_{ij})_{j=1}^m$ indicates the capabilities of site i and is reported by site i. The service vectors and service matrix may therefore include misrepresented values.

There is a *controller site* s_0 that is responsible for composing and orchestrating the Web services provided by sites $s_1, \ldots, s_n$ and for distributing the resulting profit among those sites. This controller may represent an agent whose directions the participating sites have agreed to follow. For example, it may be an agent provided by a neutral trade organization. However, it is not necessary for the controller to have a physical embodiment. It may be abstract, representing, for instance, a distributed protocol that the sites agree to follow.

The controller cannot directly prevent the other sites from misrepresenting their service vectors. The controller can manipulate the situation using only two controls. First, it may decide to exclude some sites from its Web-service compositions or, more generally, vary the level of participation of each site. For example, it may decide to route all customers to store s_1 even though s_2 claims it can provide better service. Second, it may choose the manner in which the profit resulting from successful transactions is distributed among the sites. The result of the first decision is an $n \times m$ *task matrix* T whose (i, j)'th element indicates the number of times site j is selected for service i. The result of the second decision is an m-element *distribution vector* d whose j'th element indicates the fraction of the profit that is allotted to site j. Thus, the elements of d must sum to 1. The controller makes these decisions based solely on the service matrix E. That is, both t and d are functions of E and we write them as $t(E)$ and $d(E)$. We refer to the pair of functions (t, d) as the *controller strategy*.

Our main tool is the idea of an incentive-compatible reward scheme: We seek controller strategies such that no site will benefit from misrepresenting its service vector. The controller strategy is assumed to be known to all sites. In other words, we do not consider potential solutions in which a site cannot misrepresent itself because it does not know the method used by the controller to determine the task matrix and distribution vector. Such solutions would be difficult to apply, especially when the controller represents merely a protocol used by the sites. We use $\hat{E}$ to denote the true effectiveness matrix, consisting of the accurate effectiveness values $\hat{e}_{ij}$ for all sites i and service j (and similarly for the service vector $\hat{v}$). For an m-element vector x, Let $E|_{i/x}$ denote the result of replacing the i'th row of E with x. We may then state the desired property as follows, using the notation d_i for the i'th element of vector d and, by slight abuse of notation, using $T(E)$ to denote the overall value produced by a configuration operating according to the task matrix $T(E)$:

$$\forall i \forall E : d_i(E|_{i/\hat{v}_i}) T(E|_{i/\hat{v}_i}) \geq d_i(E) T(E) \qquad (2)$$

Informally, the above condition states that in order to maximize its expected payoff, every site i cannot do any better than report its true effectiveness vector, irrespective of what other sites report (in particular, irrespective of whether they misrepresent themselves).

As a simple illustration of the above ideas, let us consider a modified version of the scenario of Figure 1 in which we hide the structural details of the example, such as the click-through rates on individual links. Our development follows the general approach outlined by Dutta and Barbera [14]. There are two key services, store and merchant, provided by a collection of n sites. Henceforth, we shall refer to the store service (attracting and routing customers) as the first service and to the merchant service (fulfilling orders) as the second service. To simplify the discussion, we assume that the sites of each kind are homogeneous. Thus all stores have identical effectiveness vectors $v_s = (x_1, x_2)$ and similarly for merchants: $v_m = (y_1, y_2)$. Let us further assume that the store service is intuitively more difficult than the merchant service. (This assumption is not unreasonable if one assumes that the merchants are selling products that, by their nature, do not require any special handling or care and that, as is often the case, attracting customers is difficult.) In particular, we assume that, on the store service, store sites are more effective than merchant sites but that, on the merchant service, they are equally effective. That is, $x_1 > y_1$ and $x_2 = y_2$. For concreteness, let $x_1 > x_2 = y_2 > y_1$.

Suppose the controller uses a simple rule for determining the task matrix: All sites that claim to be stores are assigned to the store service and the rest are assigned to the merchant service. (This rule does not bother to avoid situations resulting in no stores or no merchants. Such refinements are easily modeled using the vectors l and u introduced earlier, but are omitted for brevity.) The overall effectiveness of the system can now be expressed as a function of the number s_s of stores assigned to the store task and the number s_m of merchants assigned to the store task. (Since $x_2 = y_2$, the composition of the $n - s_s - s_m$ sites assigned to the merchant task is immaterial.) In the following, as in Equation 2, we abuse notation slightly and use $T(\cdot)$ to mean the overall value (effectiveness) of a configuration that operates according to the task matrix $T(\cdot)$.

$$T(u_s, u_m) = x_1 u_s + y_1 u_m + x_2(n - u_s - u_m) \qquad (3)$$

Now consider the following distribution vector:

$$\begin{aligned} d(u_s, u_m) &= (d_1, d_2) \quad \text{where} \qquad (4) \\ d_1 &= \frac{1}{n} + \frac{x_1(n - u_s)}{u_s t(u_s, u_m)} \quad \text{and} \\ d_2 &= \frac{1}{n} - \frac{x_2}{T(u_s, u_m)} \end{aligned}$$

Using $(x_1, x_2, y_1) = (8, 4, 2)$ and a case analysis of our small example with five sites, we can verify that the above satisfies Equation 2, i.e., no site gains by misrepresenting its capabilities. Further, we can show that, under a modest set of assumptions, this property is true for all values of the input parameters, not only those used in our example.

IV. STABILITY

In our problem formulations so far, we have focused on only the benefits resulting from the cooperation among sites offering Web services, and ignored the costs. In this section, we introduce these costs and study their ramifications, focusing on the issue of stability.

When a site enters into Web-service agreements with other sites, it may incur costs of various kinds. In our stores-and-merchants example of Section II, a participating store incurs the disk, CPU, and network costs of indexing additional data, as well as the personnel costs due to added complexity. Similar costs are also incurred by a participating merchant. In addition, a merchant may wish to assess the business cost of losing some control over its sales channel. As is the case for benefits, the precise semantics assigned to the notion of a cost of participation (financial cost, disk space, network bandwidth, or some combination of these and other factors) is immaterial to our approach. We are interested

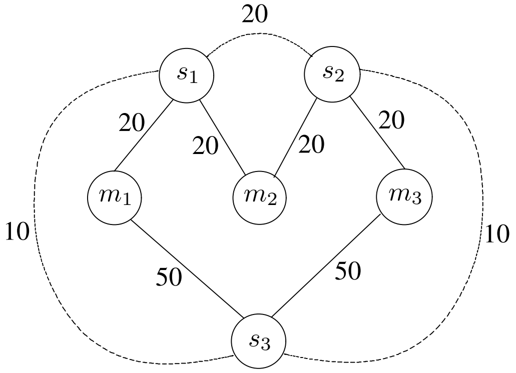

Fig. 3. The example of Figure 1 with an additional store, s_3, and possible inter-store links (dashed lines). The number adorning each link represents its cost.

in understanding how such costs affects sites' strategies for Web-service agreements, and especially how they affect the stability of agreements.

Let us augment the example of Figure 1 by introducing an additional store, s_3, as suggested by Figure 3. Further, we now permit links not only between store and merchants, but also between two stores, as suggested by the dashed lines in the figure. If two stores are connected using a link of the latter kind, it means each lists and sells products from the other using a cross-listing agreement. For concreteness, we assume the profit from such sales is divided equally between the two stores, although our method is also applicable to other schemes for dividing profits. The cost of each link is indicated by the number next to the corresponding line in the figure.

Using our calculations of Section II, we may verify that the inter-store and store-merchant link costs summarized by Figure 3 are low enough in comparison with the expected profits summarized by Table III that if some store is not connected, perhaps indirectly, via another store, to some merchant, then that store can improve its expected payoff by forming links to ensure such a connection. Thus, we may henceforth ignore the profit calculations for our example and focus only on minimizing link costs, given the need for each store to connect, directly or indirectly, with each merchant.

Consider the link-network suggested by the diagram labeled T in Figure 4. It differs from the one suggested by Figure 1 due to the link between stores s_1 and s_2. Now suppose a new store, s_3, appears and forms links with merchants m_1 and m_3, as suggested by the diagram labeled P. We use dashed lines to indicate newly formed links, such as the links (s_3, m_1) and (s_3, m_3).

Store s_1 now has the option of using the path (s_1, s_3, m_3) to connect to merchant m_3 via store s_3 instead of the path (s_1, s_2, m_3) that is currently in use. Given the costs from Figure 3, the cost of the new link (s_1, s_3) required to enable the path (s_1, s_3, m_3) is lower than that of link (s_1, s_2), which may be removed since it is no longer necessary for s_1 to connect to m_3. By symmetry, it follows that store s_2 also prefers link (s_2, s_3) to (s_1, s_2). As a result, the network transits to the configuration Q. In any configuration, we use dotted lines with an X mark on them, such as the non-link (s_1, s_2) in configuration Q, to suggest the absence of links that existed in the previous configuration.

In configuration Q, store s_3 may connect with merchants m_1 and m_2 using paths (s_3, s_1, m_1) and (s_3, s_2, m_3). Thus the direct links (s_3, m_1) and (s_3, m_3) incur unnecessary costs and are deleted, resulting in configuration R. In R, however, the links (s_1, s_3) and (s_2, s_3) no longer provide stores s_1 and s_3 the connections to m_1 and m_3. As a result, there is no incentive for s_1 and s_3 to incur the links' costs and they are removed. In order to maintain connectivity with all merchants, the link (s_1, s_2) is reinstated, resulting in a return to configuration P. This cycle of transitions repeats indefinitely and the system fails to reach a stable state.

The stability of a configuration depends not only on the salient parameters of the problems, such as the link costs in our example, but also on the rules governing transitions between configurations. The example of Figure 4 implicitly assumed a model that permits a site s_i to unilaterally create a link (s_i, s_j) if it is willing to bear the link's cost. That is, the possibility that site s_j may refuse such a link, even though it costs it nothing, is not considered. (Given our setup, this assumption is reasonable but it is easy to imagine alternate setups in which it fails.) Similarly, our example implicitly assumed that a site may unilaterally sever a link. Transitions typically have preconditions that a configuration must satisfy before the transition can be taken. In our example, the add-link transition has the simple precondition that the link's addition must result in a net benefit to one of its end-points. In general, therefore, we say a configuration is stable with respect to a set of transitions if there is no transition in this set that has its preconditions satisfied. Such analysis of the stability of a a Web-services environment sometimes yields a concise characterization of stable configurations. More generally, such analysis reveals scenarios that admit stable configurations and can serve as a guide for the creation and evolution of Web-service agreements among sites.

V. RELATED WORK

A compelling argument for distributed computation and, by extension, Web services, based on the costs of moving data and the costs of computation is presented by Gray [21]. There are several examples of popular, albeit simple, Web services, such as Amazon Web Services [7] and the collection of Google APIs [19]. The Amazon Mechanical Turk [1] is an interesting example of a Web service that uses human effort to a large degree.

Several initiatives focus on providing methods for dynamic interoperation of services, including standards such as Java Message Service [25] and UDDI [32]. UDDI enhancements for building registries based on quality of service are reported by Kumar et al. [28]. Technologies such as SOAP [30], [22], [23], WSDL [13], [13], BPEL4WS [2], and Enterprise Java Beans [27], [3] address different parts of the interoperation puzzle by providing a common framework for specifying properties, orchestrating services, etc.

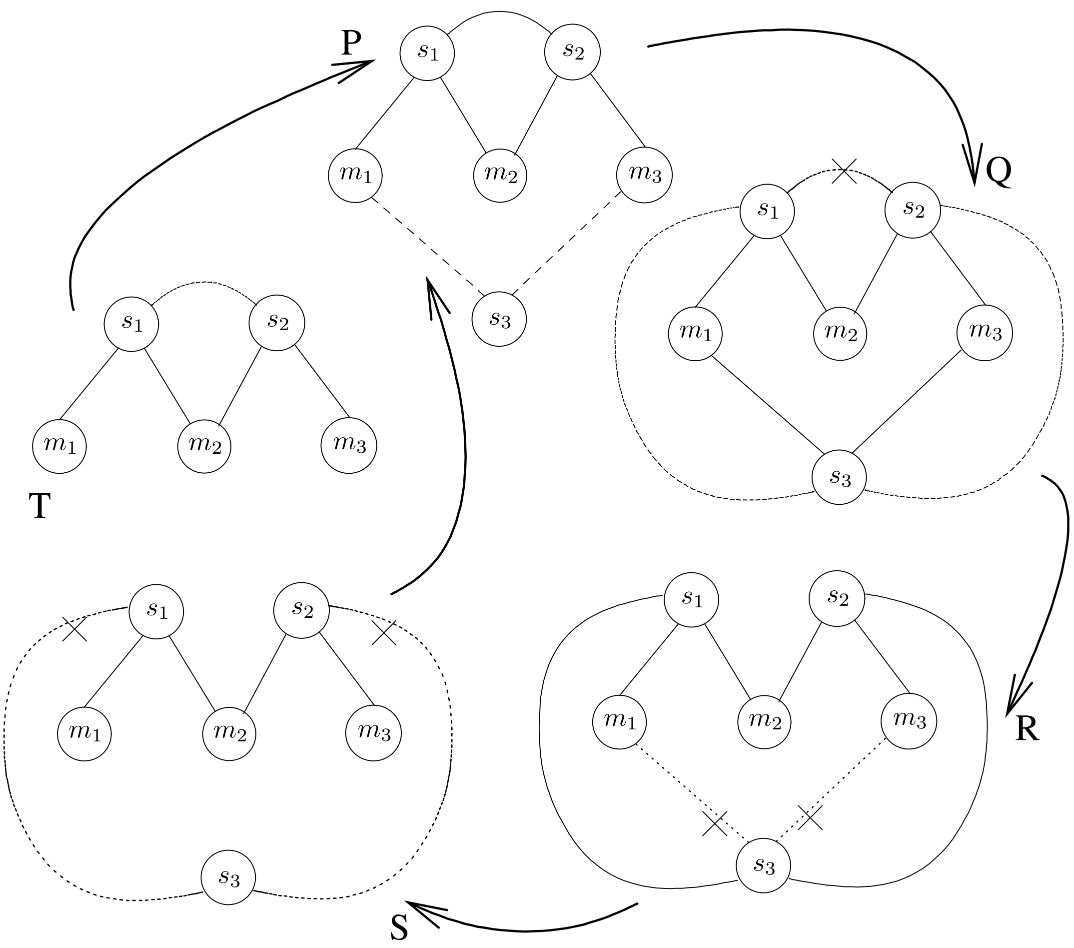

Fig. 4. The evolution of inter-site agreements for the network of Figure 3 depicted as a collection of configurations, labeled $P, \ldots, T$. Each arrow denotes a transition from one configuration to another.

OWL-S and related work [29], [18] addresses the interaction between Web services and the real world, and the semi-automatic composition of Web services [36], [35]. The so-called Roman model focuses on abstract activities modeled using finite-state automata. This model is extended in *Colombo* to include message passing, atomic processes, and a model of the real world as a relational database, leading to techniques for automatic composition [9], [8]. Arsanjani et al. describe methods for representing service semantics to enable automatic composition [4], [6], [5].

The problem of trust among sites is closely related to the concerns addressed in this paper. The problem of effectively collaborating with untrusted or partly untrusted parties has been addressed by methods such as privacy homomorphisms [11] and queries on encrypted data [24]. In a completely trusted setting, issues such as stability are still important, although others such as misrepresentation may disappear.

Our methods in this paper draw on concepts first developed in the game theory and economics literature. As noted earlier, the scheme for distributing profits described in Section II is based on the well-known Shapley value [34]. Our framework for casting strategic issues in Web-service composition in game theoretic terms opens the door for the application of classic results on bargaining, including those by Nash [31] and Kalai and Smorodinsky [26]. In earlier work, we have explored issues of stability and fairness in a transportation setting [12]. Our treatment of misrepresentation draws on Dutta and Barbera's work on avoiding adverse selection [14]. The form of stability presented in Section IV is a first step toward exploring stronger forms of stability [15], [33]. In this respect, Bondareva's balancedness result is notable because it permits characterization of solutions in many situations that do not yield to other methods [10].

VI. CONCLUSION

We motivated the need to study the strategic aspects of Web-service composition and discussed three specific problems in this area. We cast the problem of distributing the profits resulting from a collaboration of Web-service providers (sites) in a game theoretic framework and used the Shapley value to illustrate schemes for fairly distributing the profits of service composition over the participating sites. We used the idea of incentive-compatible reward schemes to illustrate how agents (human or artificial) responsible for composing Web services can avoid potential problems resulting from services that overstate their capabilities. We briefly illustrated unstable configurations that may result when each site enters into agreements with other sites guided by the goal of maximizing its expected profit.

In continuing work, we are determining ways to characterize Web-service scenarios that admit clean and efficient solutions to the three problems discussed in this paper, as well as other related problems. For example, we are working on characterizing scenarios that guarantee stable solutions while allowing the use of incentive-compatible reward schemes. We are also studying a variant of the problem addressed in Section III in which a site reports not only its own capabilities but also those of others. Now a site may not only misrepresent its own capabilities, but also misrepresent those of others. On the other hand, the multiplicity of reports on a site's capabilities provides additional opportunities for devising incentive-compatible schemes.

REFERENCES

[1] Amazon Mechanical Turk. `http://www.mturk.com/`, 2005.

[2] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. Business process execution language for web services (BPEL4WS)–version 1.1. `http://www.ibm.com/developerworks/library/ws-bpel/`, 2004.

[3] Ali Arsanjani. Enterprise component services. *Communications of the ACM*, pages 30–34, October 2002.

[4] Ali Arsanjani. Explicit representation of service semantics: Towards automated composition through a dynamically re-configurable architectural style for on demand computing. In *Proceedings of the IEEE International Conference on Web Services (ICWS)*, pages 34–37, Las Vegas, Nevada, June 2003.

[5] Ali Arsanjani and James J. Alpigini. Using grammar-oriented object design to seamlessly map business models to software architectures. In *Proceedings of the IASTED Conference on Modeling and Simulation*, Pittsburgh, Pennsylvania, 2001.

[6] Ali Arsanjani, Francisco Curbera, and Nirmal Mukhi. Manners externalize semantics for on-demand composition of context-aware services. In *Proceedings of the IEEE International Conference on Web Services (ICWS)*, pages 583–590, San Diego, California, July 2004.

[7] Amazon.com Web Services 2.0 API and integration guide. Amazon.com Web Services Documentation. `http://associates.amazon.com/`, January 2003.

[8] Daniela Berardi, Diego Calvanese, Giuseppe De Giacomo, Richard Hull, and Massimo Mecella. Automatic composition of transition-based semantic web services with messaging. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 613–624, Trondheim, Norway, August 2005.

[9] Daniela Berardi, Diego Calvanese, Giuseppe De Giacomo, Massimo Mecella, and Rick Hull. Automatic composition of web services in Colombo. In *Proceedings of the 13th Italian Conference on Database Systems (SEBD)*, pages 8–15, 2005.

[10] Olga Bondareva. Kernel theory in N-person games. *Vestnik Leningrad Univ. Math*, 17:141–142, 1962.

[11] Claus Boyens and Matthias Fischmann. Profiting from untrusted parties in Web-based applications. In *Proceedings of the 4th International Conference on E-Commerce and Web Technologies*, volume 2738 of *Lecture Notes in Computer Science*, pages 216–226, Prague, Czech Republic, September 2003. Springer-Verlag Heidelberg.

[12] Sudarshan S. Chawathe. Fair policies for travel on neighborhood streets. In *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1027–1032, Vienna, Austria, September 2005.

[13] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web services description language (WSDL) 1.1. W3C Note. `http://www.w3.org/TR/wsdl/`, March 2001.

[14] Bhaskar Dutta and Salvador Barbera. Incentive compatible reward schemes for labour-managed firms. *Review of Economic Design*, 5(2):111–127, 2000.

[15] Francis Ysidro Edgeworth. *Mathematical Psychics*. Kegan Paul, 1881.

[16] fedex.com terms of use. `http://www.fedex.com/global_legal.cgi?cc=us&sc=/legal/`, February 2006.

[17] David Flater. XTide. `http://www.flaterco.com/xtide/`, 2005.

[18] Thibaud Flury, Gilles Privat, and Fano Ramparany. OWL-based location ontology for context-aware services. In *Proceedings of the Workshop on Artificial Intelligence in Mobile Systems*, Nottingham, U.K., September 2004.

[19] Google APIs. `http://code.google.com/apis.html`, 2005.

[20] Froogle. `http://froogle.google.com/`, 2006.

[21] Jim Gray. Distributed computing economics. In Andrew Herbert and Karen Sparck Jones, editors, *Computer Systems: Papers for Roger Needham*, pages 77–82, Cambridge, England, February 2003. Microsoft Research Limited.

[22] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen. SOAP version 1.2 part 1: Messaging framework. W3C Recommendation. `http://www.w3.org/TR/soap12-part1/`, June 2003.

[23] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen. SOAP version 1.2 part 1: Messaging framework. W3C Recommendation. `http://www.w3.org/TR/soap12-part2/`, June 2003.

[24] Hakan Hacigumus, Bala Iyer, Chen Li, and Sharad Mehrotra. Executing SQL over encrypted data in the database-service-provider model. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 216–227, 2002.

[25] Mark Hapner, Rich Burridge, Rahul Sharma, Joseph Fialli, and Kate Stout. Java Message Service. Specification. Sun Microsystems. `http://www.sun.com/`, April 2002.

[26] Ehud Kalai and Meir Smorodinsky. Other solutions to Nash's bargaining problem. *Econometrica*, 43:513–518, 1975.

[27] Anup Kumar. Distributed system development using web service and Enterprise Java Beans. In *Proceedings of the IEEE International Conference on Services Computing (SCC)*, Orlando, Florida, July 2005.

[28] Anup Kumar, Ayman El-Geniedy, and Sanjuli Agarwal. A generalized framework for providing qos based registry in service oriented architecture. In *Proceedings of the IEEE International Conference on Services Computing (SCC)*, pages 295–306, Orlando, Florida, July 2005.

[29] David Martin, Massimo Paolucci, Sheila McIlraith, Mark Burstein, Drew McDermott, Deborah McGuinness, Bijan Parsia, Terry Payne, Marta Sabou, Monika Solanki, Naveen Srinivasan, and Katia Sycara. Bringing semantics to web services: The OWL-S approach. In *Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC)*, 2004.

[30] Nilo Mitra. SOAP version 1.2 part 0: Primer. W3C Recommendation. `http://www.w3.org/TR/soap12-part0/`, June 2003.

[31] John Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[32] Introduction to UDDI: Important features and functional concepts. White paper, Organization for the Advancement of Structured Information Standards (OASIS). `http://www.uddi.org/`, October 2004.

[33] Herbert E. Scarf. The core of an N-person game. *Econometrica*, 35:50–69, 1967.

[34] Lloyd S. Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, volume 28, pages 307–312. Annals of Mathematical Studies. Princeton University Press, 1953.

[35] Mithun Sheshagiri. Automatic composition and invocation of semantic web services. Master's thesis, University of Maryland, Baltimore County, August 2004.

[36] Evren Sirin, James Hendler, and Bijan Parsia. Semi-automatic composition of web services using semantic descriptions. In *Proceedings of the Web Services: Modeling, Architecture and Infrastructure Workshop*, Angers, France, April 2003. Workshop associated with the International Conference on Enterprise Information Systems.

[37] UPS tracking terms and conditions. `http://wwwapps.ups.com/WebTracking/terms?loc=en_US`, February 2006.